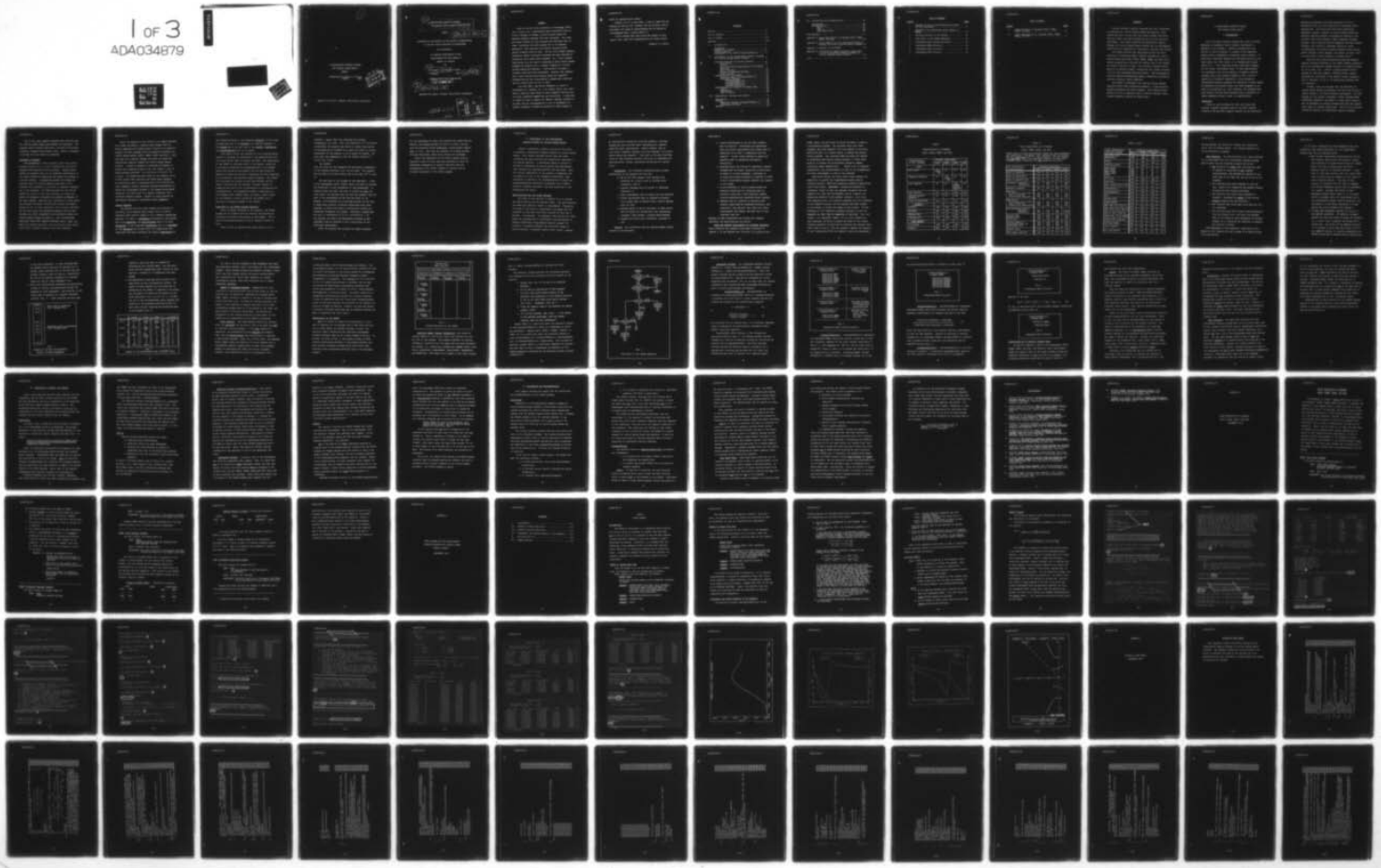
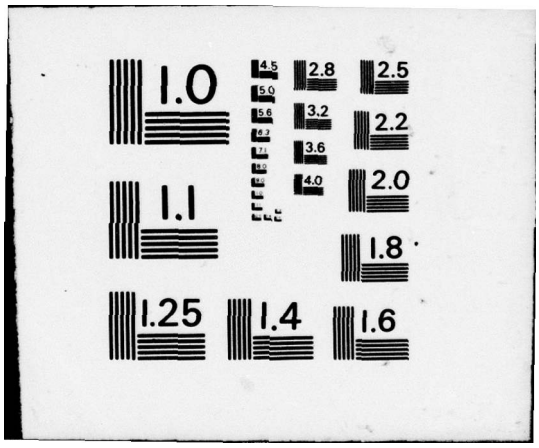
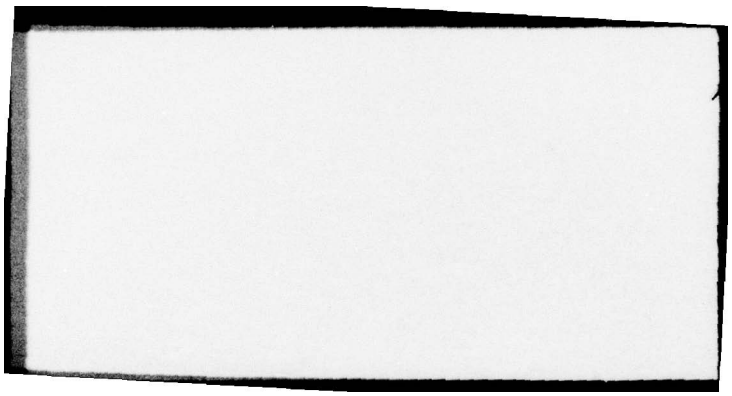
AD-A034 879 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPC--ETC(U)
DEC 76 F L O'BRIEN

UNCLASSIFIED GE/EE/76D-38 NL

1 OF 3
AD A034879




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6



(1)

A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN

THESIS

GE/EE/76D-38 Frederic L. O'Brien
Captain USAF

Approved for public release; distribution unlimited.

(6)

A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPCSD).

THESIS (9) Master's thesis,

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

(10) Frederic L. O'Brien B.S.E.E.

Captain USAF

(12) 219 p.

Graduate Electrical Engineering

(11) December 1976

(14) GE/EE/76D-38

Approved for public release; distribution unlimited.

012225

LB

Preface

While at the Air Force Institute of Technology (AFIT), other students and I experienced sheer frustration when we tried to analyze or design a control system using the separate computer programs that were available. The input and output formats for each program were different and, at times confusing, and each program had to be attached separately, then executed, and finally returned before another program could be used. The Air Force Flight Dynamics Laboratory (AFFDL) personnel expressed the same feelings of frustration when using these programs. So, I have plunged head-strong into the task of combining as many control design programs as possible into a "usable" program for control systems design and analysis. Given another two years, I probably could have been successful. However, the comments that I have received from persons using the "separate" programs and the new "consolidated" program have indicated that my efforts were not all in vain.

I am sure that I had become somewhat of a pest to Professors C.W. Richard, Jr., T.E. Reeves, and G. Orr, and I wish to publicly thank them for their patience with me, and for their competent suggestions and directions. I wish also to thank Dr. Gary B. Lamont for taking a special interest in my task, and for encouraging me to try to accomplish it. Similar thoughts of thanks go directly to Frank George of

AFFDL for sponsoring my thesis.

Finally, but in no way least, I wish to thank God for helping me, and my wife, Barbara, and my children, Noelle and Kathy, for being so understanding and for giving me encouragement when I really needed it.

I will confess that some errors may appear in this report, and I take full responsibility for these errors.

Frederic L. O'Brien

Contents

List of Figures

List of Tables

ABSTRACT

Several different computer programs have been independently developed for control system design and analysis. Each program may have its own input and output formats (different from any other), and each program may have its own operating philosophy. This report describes a Consolidated Computer Program for Control System Design (CCPCSD) and its development: specifications, organization, realization, testing, and results.

The CCPCSD contains four previously written Air Force Institute of Technology (AFIT) computer programs for control system design algorithms (ROOTL, PARTL, FREQR, and POLY) and a lead-in program which controls the overall flow of the CCPCSD. An overlay structure is used. The structure consists principally of an executive (main) overlay and four primary overlays (one for each subprogram mentioned above). Each subprogram is modified towards standardizing inputs, termination procedures, and data structure (where feasible).

The CCPCSD is developed to be used by control systems engineers and/or AFIT engineering students. A user-oriented approach permeates the entire program. This computer-aided design tool can be operated from an intercom terminal, where operator-computer interaction takes place.

A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN

I. Introduction

One of the most important present-day tools for either designing or analyzing control systems (continuous or digital) is the computer. Computer programs have been written to aid the engineer or student in the sometimes awesome task of designing or analyzing control systems. It seems that individual programs are willingly written as the need arises, but with little, or no thought about future needs (i.e., ...for combining several individual programs for one large, consolidated task...or more specifically, for combining a root locus program, a frequency response program, and a time response program so that a control system can be analyzed completely.). Well structured consolidated computer programs are definitely in great demand by engineers and students alike, but these programs do not seem to be available (or, quite possibly, the programs have been written, but are made available only to personnel in small unexposed offices, firms, or organizations).

Background

Prior to, and including the time this thesis was written, computer programs stored in the AFIT computer library of the CDC 6600 computer system, and the Electrical

Engineering Department were made available for use of personnel at the Air Force Institute of Technology (AFIT), building 640. The library and Electrical Engineering Department possess a variety of control system design and analysis computer programs (continuous and digital), such as ROOTL (Root Locus Program), FREQR (Frequency Response Program), PARTL (Partial Fraction Expansion and Time Response Program), POLY (Roots of Polynomial Function), DIGIS (Z-Transforms), FIR (Finite Impulse Response Filter Design), OPTCON (Optimal Control Design), and others, no doubt, that are still part of some professor's "private stock."

There are also other specialized design and analysis computer programs available at the Flight Dynamics Laboratory and at the Aeronautical Systems Division of Wright-Patterson Air Force Base, Ohio. DIGICON ("Digital Flight Control Systems for Tactical Fighter," AFFDL-TR-74-69; computer algorithm for the design of the digital control systems of the F-4 Aircraft) is one of the specialized programs that are available.

Further, there are programs that are available for control system design and analysis that have been written to fulfill master's, or doctor's degree requirements. One such program is "The Design of Automatic Control Systems Using Interactive Graphics," by Kenneth R. Young, which allows a user an advantage of working interactively with the computer (real time) for control system work. The Air Force has also contracted industry for specialized computer programs.

All in all, many computer programs that could be used for control system design and analysis are available. The engineer and/or student would like to be able to use these programs---computer-aided design tools. Hence, a statement of purpose for this thesis is presented.

Statement of Purpose

The professional control engineer and/or the control engineer student are, of course, saddled with the task of designing and analyzing control systems. Computer-aided design and analysis tools are an important contribution to the timely success of completing the design and analysis of a control system. To adequately perform the design and analysis task, the control engineer must have at his disposal (as a minimum) a root locus, a frequency response, and a time response. The characteristics of the control system are shown by these three methods, and these then enable the designer to analyze the complete performance of his basic system. There are also extensions to these design and analysys techniques (Nichols Chart design and Nyquist design are two of these extensions) that are also available, but the root locus, the frequency response, and the time response are still considered as the principal methods for control system design and analysis. And, as previously mentioned, three AFIT computer programs (ROOTL, FREQR, and PARTL) are already available, and they provide these basics (root locus, frequency response, and time response).

Hence, the purpose of this effort is to clearly develop, by at least one means, a computer-aided design tool for control engineers and students which contains as an absolute minimum four AFIT programs that are presently available for control system design (ROOTL, FREQR, PARTL, and POLY). This tool must be a computer program that shall be called the Consolidated Computer Program for Control System Design (CCPCSD). It must be capable of providing an engineer or student with repeated access to one, two, three, or all of these programs mentioned - all during one sitting at the terminal. The capability must exist for the operator to be in full control when selecting and using these programs. An architecture must be developed which allows adding some other programs, without requiring extensive modification to either the consolidated program or the program being added. Therefore, the CCPCSD must be a useful, versatile, self-contained computer program - capable of being operated by experienced engineers or beginning student engineers.

General Approach

The approach that was taken towards the objective of providing a good computer-aided design tool for control engineers might best be thought of as a "general engineering approach," in that: 1) information pertaining to the problem has to be researched, and specifications have to be fully determined; 2) the researched information has to be evaluated so that decisions can be made about the capabilities and trade-offs that would influence the overall organization of

the problem solution; 3) the organized solution, in this case an algorithm, has to be realized in a computer program; 4) the program has to be test; and 5) the results, conclusions, and recommendations have to be stated.

This report covers this approach in the following manner. Chapter II includes the definition of the specifications and the actual method taken to achieve the task. The organization of the CCPCSD is discussed fully in the chapter, along with a discussion concerning the standardization of the program inputs and termination procedures, and the data structure. Chapter III shows the process of realizing the total algorithm as an efficient, error-proofed computer program. The chapter also discusses the testing criteria and methods, and the results obtained from the testing. Finally, Chapter IV summarizes the effort with a presentation of conclusions and recommendations. The appendixes contain a description of the programs ROOTL, FREQR, PARTL, and POLY; a user's guide for the CCPCSD; a source listing for the CCPCSD; and a procedure for adding programs to the CCPCSD.

Highlights of the CCPCSD Program Operation

From a terminal (teletype or CRT display), the CCPCSD program can be attached from the library, and executed by following the printed instructions as they appear. Now to look at the program operation from an operator's point of view.

First of all, an operator must state that he is at a

terminal. Typing "TTY" and returning the carriage accomplishes this task. Next, and explanation of the program is provided; the operator may select or reject this option. The explanation is intended to be used for either an initial declaration of the program, or as a recall for review. For each case, the explanation tells the general sequence of events that occurs.

The program
that the roots for the numerator and denominator polynomials of the transfer function, G(s), can be found. The operator may not need to use this option, and he can type "NO" to move on.

The next part of the program is very important. A list of the subprograms (ROOTL, FREQR, PARTL, and POLY) is printed out along with a brief explanation of each subprogram. To conserve paper and time, the list is printed out only one time; hence, the operator must jot down the symbolic name for each of the subprograms for use then and later in the program. (This decision for a one-time printout of the list is based upon previous irritating experiences with other programs, in which long lists of options were printed out repeatedly throughout the program. Hopefully, copying down the list of subprograms is not too inconvenient.) Now the operator can select the subprogram he wishes to use. (Appendix A may be referred to for a description of the individual subprograms.)

After the operator has provided the inputs required

for the subprogram he chose, and obtained the output that he desired, the program allows the option to either continue with the presently chosen subprogram, choose another subprogram, or stop. Each time the operator is finished working with any subprogram, these above options are presented.

Hence, the highlights of the CCPCSD program (from an operator's point of view) have been presented to show the sequential arrangement of specific functions that are accomplished by the CCPCSD. Chapter II follows with an in-depth discussion of the CCPCSD program.

II. Development of the Consolidated Computer Program for Control System Design

Chapter I presented a general overview of the effort to develop a consolidated computer program for control system design. This chapter expounds further upon the areas concerning the specifications and organization discussed in the general approach. A substantial amount of effort is devoted to discussing the specifications and requirements that afforded the finalized approach that was taken. Next, the overall organization of the program is explained, and the make-up of each of the overlays is described. Some discussion is devoted to the standardization of inputs and program/subprogram termination procedures, and, finally, pertinent comments relating to the data structures of the subprograms are stated.

Specifications and the Actual Approach

The theme of this section of Chapter II is to discuss the specifications and the approach taken. The specifications are based upon specific needs, or requirements. Discussing (in terms of the needs or requirements) the means that are available to meet each need allow choosing the best way to proceed - the approach. The approach, when carried out fully, meets the intended goal of "providing a unified package of computer algorithms (consolidated computer program) for in-house, or acamemic analysis and conceptual design of control systems...contained totally within itself...capable

of enabling user interaction with the computer...one-time program use with one-time input (satisfying all computer algorithm input requirements)...useful outputs, such as root locus, frequency response, and time response, for control system design and analysis." This goal is the sole directing force for this important section, and prior to discussing the specifications, certain constraints and factors are exposed below.

Constraints. The following constraints place certain restrictions on the approach and the goal:

a. use of CDC 6600 computer only; maximum core available for intercom jobs is $50{,}000_8$ words. (Reference 2 and 3)
b. computer language must be Fortran IV (Extended) (Reference 1)
c. input requirements must be simple and user-oriented.
d. output requirements must be complete and useful (in a format easy to analyze from a control systems point of view)
e. maximum amount of plots, printouts, or data stored must be within the $50{,}000_8$ word core requirement. (program + data storage = $50{,}000_8$ words maximum)
f. program must allow user interaction (terminal or crt)

Factors. The constraints that are imposed demand certain factors to be addressed.

a. A good understanding of the CDC 6600 computer must be acquired. Terminology and appropriate input and output requirements and formats must be understood to be able to communicate with the computer. Correct codes relating to syntax and semantics must be understood and applied efficiently.

b. Knowledge about the Fortran IV (Extended) computer language must be sought, along with a proficiency in using it to write programs. A knowledge of overlays and overlay structure is required in order to put a consolidated program together and conserve on core usage.

c. A good knowledge of control system design and analysis terminology and techniques must be acquired, along with a valid approach towards properly analyzing and designing control systems.

d. Research must be conducted to determine what computer algorithms are presently available, what function(s) they perform, what input and output requirements they demand, and what type of data structure they use.

Knowing now what constraints and factors were imposed initially, the specifications can unfold.

Input and Outputs Requirements for Programs Available. After reviewing the programs (algorithms) mentioned in Chapter I, it was decided that initially the programs ROOTL,

FREQR, PARTL, and POLY would be chosen as basics to make up a consolidated program. The rationale being that these programs, when used together, are very capable of providing sufficient data, in a useful format, to design and analyze control systems. And, although FREQR and ROOTL are capable of performing some digital system analysis, a "first step" would be to develop the consolidated program to handle continuous domain systems only. Tables I and II indicate, respectively, the characteristics of each of the programs, and the input requirements of each of the programs.

Also worth noting, the "first step" would consider the capability to operate the consolidated program from an intercom terminal. BATCH and interactive graphics capabilities could follow later. **Rationale** - interactive graphics is considered "ideal" in that the engineer interacts with the computer in a real-time environment, and he "creates" his design and analysis at the graphics terminal. Root locus, time response plots, and frequency response plots are displayed on the graphics console, and the engineer may make changes to his system, or accept the design he has created. The key advantage to the interactive graphics approach is that the engineer can "see" what is happening in real-time. But, even though it is considered "ideal," it is a very complex and time consuming job which is beyond the scope of this thesis. Intercom operation is preferred by engineers and students over BATCH jobs, in that it, like the graphics, enables the operator to act interactively with the computer (input the parameters,

Table I

Characteristics of Programs

ROOTL, FREQR, PARTL, and POLY

Characteristics of each Program	Programs (Algorithms)			
	ROOTL	FREQR	PARTL	POLY
...Job...				
Root Locus	yes (order 50)	no	no	no
Frequency Response	no	yes (order 20)	no	no
Time Response	no	no	yes (order Num:12 Den:24)	no
Time Function	no	no	yes	no
Partial Fraction Expansion	no	no	yes	no
Roots of polynomial	yes	no	no	yes
...Outputs...				
Listing	yes	yes	yes	yes
TTY Plot	no	yes	yes	N/A
Printer Plot	no	no	yes	N/A
Calcomp Plot	yes	yes	yes	N/A
...Type System...				
Continuous	yes	yes	yes	N/A
Digital	part.	yes	no	N/A

Table II

*Input Requirements for Programs

ROOTL, FREQR, PARTL, and POLY

*NOTE: Table II shows the different "ways" the input parameters are "asked for." The symbol "N/A" indicates that the program does not explicitly "ask" for that parameter, but the program may implicitly require the parameter. Hence, at times three different ways of entering the same parameter may occur.

Input Requirements For Each Program	Programs (Algorithms)			
	ROOTL	FREQR	PARTL	POLY
Order of Numerator	N/A	yes	yes	N/A
Order of Denominator	N/A	yes	yes	N/A
Power of Repeated Root	N/A	no	yes	N/A
Polynomial: Numerator	no	yes	no	yes
Denominator	no	yes	no	yes
Numerator Constant	yes	yes	yes	N/A
Numerator:				
No. Factors Given Order	N/A	yes	N/A	N/A
Order of Factors	N/A	yes	N/A	N/A
Coefficients Numerator	N/A	yes	N/A	N/A
Denominator:				
No. Factors Given Order	N/A	yes	N/A	N/A
Order of Factors	N/A	yes	N/A	N/A
Coefficients Denominator	N/A	yes	N/A	N/A
Real/Imaginary Factors:				
Numerator	N/A	N/A	yes	roots
Denominator	N/A	N/A	yes	roots
Initial Time	N/A	N/A	yes	N/A
Duration Time	N/A	N/A	yes	N/A
Delta Time	N/A	N/A	yes	N/A
Initial Omega (w)	N/A	yes	N/A	N/A
Incremental Omega (w)	N/A	yes	N/A	N/A
Final Omega (w)	N/A	yes	N/A	N/A
Restart Array	N/A	yes	N/A	N/A
Time Delay, exp(-Ts)	yes	yes	no	N/A

Table II cont.

Input Requirements For Each Program	Programs (Algorithms)			
	ROOTL	FREQR	PARTL	POLY
Magnitude Plot Log-dB Plot	N/A N/A	yes yes	N/A N/A	N/A N/A
Titles	yes	yes	yes	N/A
Sampling Period	no	yes	no	N/A
Right Bound Left Bound Upper Bound Lower Bound Bound	yes yes yes yes yes	no no no no no	no no no no no	N/A N/A N/A N/A N/A
Plot Size	yes	yes	yes	N/A
Step Size	yes	N/A	N/A	N/A
Dist. Between Plot. Pts.	yes	no	no	N/A
Sensitivity, K	yes	N/A	N/A	N/A
Gain Bandwidth	yes	N/A	N/A	N/A
Zeta	yes	N/A	N/A	N/A
Initial Step	yes	N/A	N/A	N/A
Figure Number	yes	no	no	N/A
Real Part Init.Test Pt.	yes	N/A	N/A	N/A
Imag. Part Init. Test Pt.	yes	N/A	N/A	N/A
Type of Init. Test Pt.	yes	N/A	N/A	N/A
Number Pts. to be Called	yes	N/A	N/A	N/A
Delete Tnsfr Function	yes	no	no	N/A
Multiple Graphs	yes	no	no	N/A
Plots Disposed	yes	yes	yes	N/A
Plot Destination	yes	yes	yes	N/A

see and analyze the printed out results, and continue or stop), but to a lesser degree. The intercom operation is not as difficult to program.

Data Structure. The specifications for a data structure to be incorporated into the consolidated computer program must be such that the following criteria can be met:

a. One structure for input data; the structure must be capable of being used by all programs simultaneously, thus allowing an operator to put input data in one time, but use any or all of the programs.

b. The structure must allow changing of only the data wished to be changed by the operator without altering the data entered previously.

c. The structure must allow future addition of programs - extending the amount of data stored without changing the data structure.

d. The structure must allow data to be read out, but not destroyed.

e. The structure must allow certain output parameters from one program (i.e., the roots from program POLY) to be stored as input parameters for another program (i.e., stored as the numerator or denominator factors of ROOTL).

Data management is very important, especially as the amount of core required for each program to be added becomes a critical factor.

Of the basic techniques for data management that were researched, four seemed applicable for the consolidated program, and they are discussed below.

a. Mass storage input/output subroutines for the CDC 6600 computer system allow the creation, access, and modification of multi-record files on a random basis, and does not depend upon the file's physical position or internal structure. A good point is that each record in the file, when either read or written into at random, does not have its other contents affected. This type of file storage is consistent with the idea of changing only certain parameters rather than all parameters of a system. Although this method of data management seemed almost ideal, it is not necessarily an optimum choice for input/output use in the consolidated program (could be tailored), and furthermore, it was nearly impossible to locate persons who would recommend this approach.

b. A second data management approach is the use of the NAMELIST statement. The NAMELIST statement permits the input and output of groups of variables (i.e., poles, zeros, constant) and arrays with a unique name. Key points of the NAMELIST data are: (1) a variable or array name may belong to one or more NAMELIST groups, (2) specific parameters in a NAMELIST group may be changed without altering any

of the other parameters, (3) when entering data, the NAMELIST group name must be preceded by a $ (dollar sign), and after all of the data has been entered, another $ must be used as a termination. The use of the NAMELIST statement is a familiar method for use as a data management tool.

c. Where only single-valued parameters are manipulated, strings of vectors (one-dimensional arrays) are adequate for data management. Each position in the vector must be assigned as a specific parameter location (Fig. 1). These positions can have data

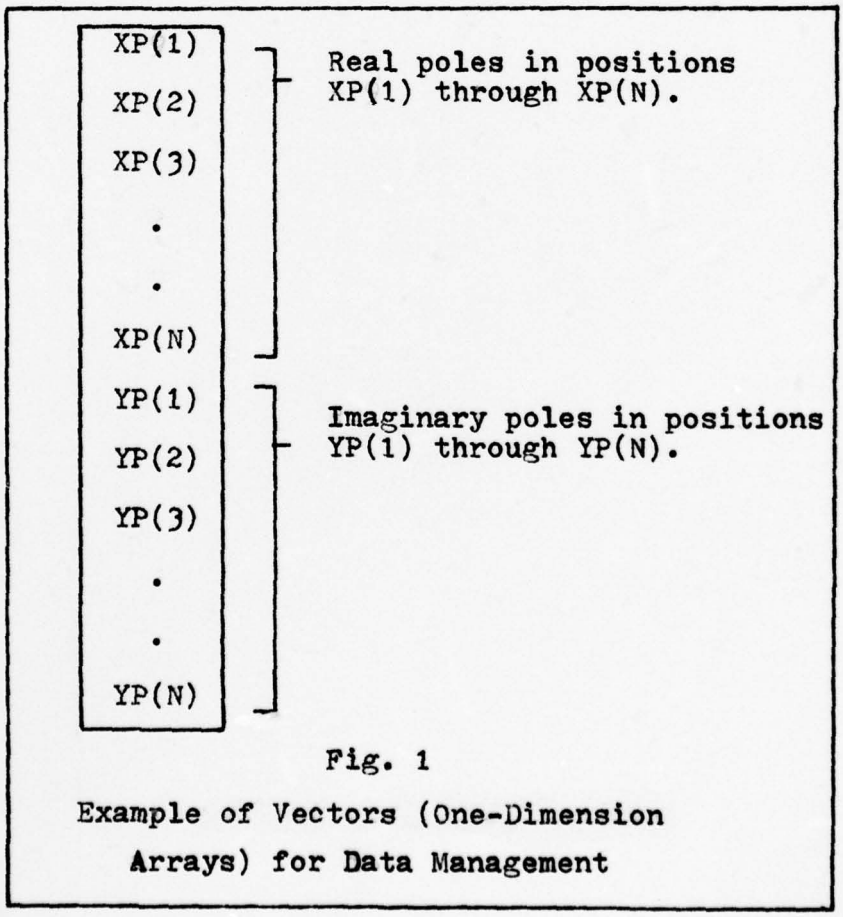


Fig. 1
Example of Vectors (One-Dimension Arrays) for Data Management

(specific) read into them, or removed for calculations and returned again. The key point being that the program must keep "record" of each position - putting in, or taking out only when needed.

d. Two-dimensional arrays may also be used in the same manner as the one-dimensional arrays. The number of words reserved for the array will be determined by the number of elements in the array, and the number is figured by the product of the array subscripts (i.e., A(3,3)= 3 x 3 = 9 words). The use of the two-dimensional array requires pre-planning of the parameter positions as well as exact data control (entering and retrieving data) by the program (Fig. 2).

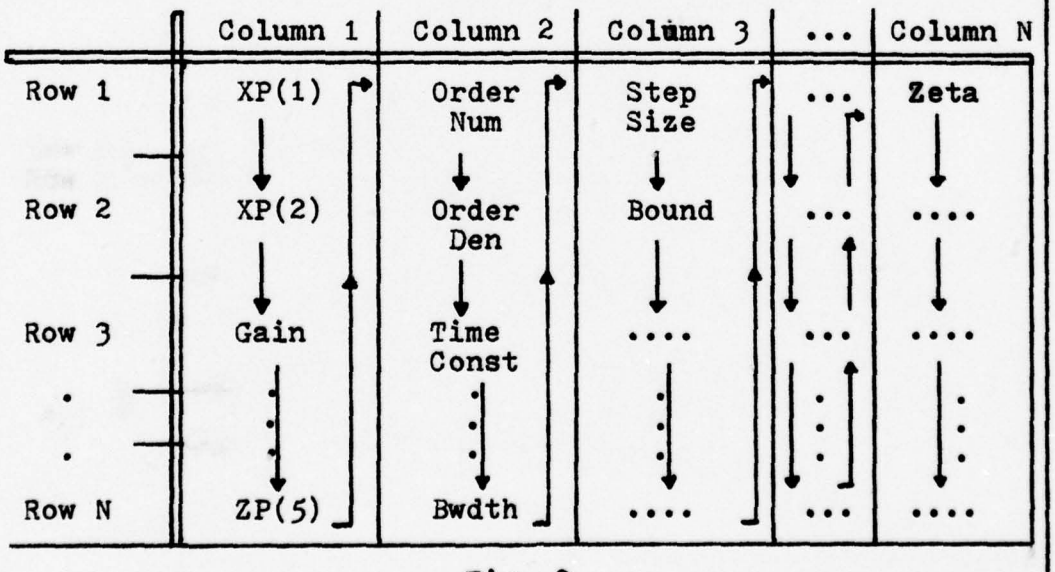


Fig. 2
Example of Two-Dimensional Array (Order of Entry →)

In view of the four methods of data management that have been described, methods b and c are chosen for the consolidated program - ROOTL already utilizes the NAMELIST statement; FREQR, PARTL, and POLY already utilize the vector storage method. And, considering the "first Step" approach to a usable consolidated program, these data structures are to remain relatively unchanged.

Method of Combining Programs. Remembering the chief constraint of having available only $50{,}000_8$ words of core for intercom use, the method for combining the programs ROOTL, FREQR, PARTL, and POLY is limited to the use of overlays and overlay structure only. By using the overlay structure, the amount of core required is reduced (only part of the total consolidated program would be in the memory at one time), and field length is used more efficiently. The details and explanation of the overlay structure is given later in this chapter as the organization of the CCPCSD is discussed.

Final Approach. The final approach takes into consideration the knowledge (of the writer) that is at hand, the goal to "develop a unified package...," the time element for completion, the availability of the computer and computer time, and the specifications that are pertinent to describe a consolidated program. Hence, as a "first step," the programs ROOTL, FREQR, PARTL, and POLY are to be put together into a consolidated computer program which is then, as a minimum, capable of performing each task (root locus, frequency response, time response and roots of a polynimial) for a

continuous domain control system design and analysis. The consolidated program is to be user-oriented, allowing the user to control the destiny of the control system he is designing and analyzing by choosing whichever program he wishes (repeatedly, if necessary). The programs are to be modified to fit in a usable architecture of overlays, and the data structures and input/output requirements presently used within each program are to remain in effect, unless glaring discrepancies, or conflicts arise. The major intent is to standardize operations and instructions wherever possible. It must be understood that the "first step" finished product shall be definitely useful, but certainly not as optimal of a solution as might be desired. However, the time and efforts of another interested individual may be channeled towards the goal of optimizing the "first step."

Organization of the CCPCSD

Again, to strive to reduce the actual amount of core that is required for the program, and to make more efficient use of field length, an overlay structure is used. The executive (main) overlay and the individual subprograms (primary overlays) make up the overall program structure. In general, as shown by Fig. 3, the primary overlay of ROOTL contains subroutines pertinent to the primary overlay, and also contains secondary overlays which have within their structure other subroutines pertinent only to the secondary overlay.

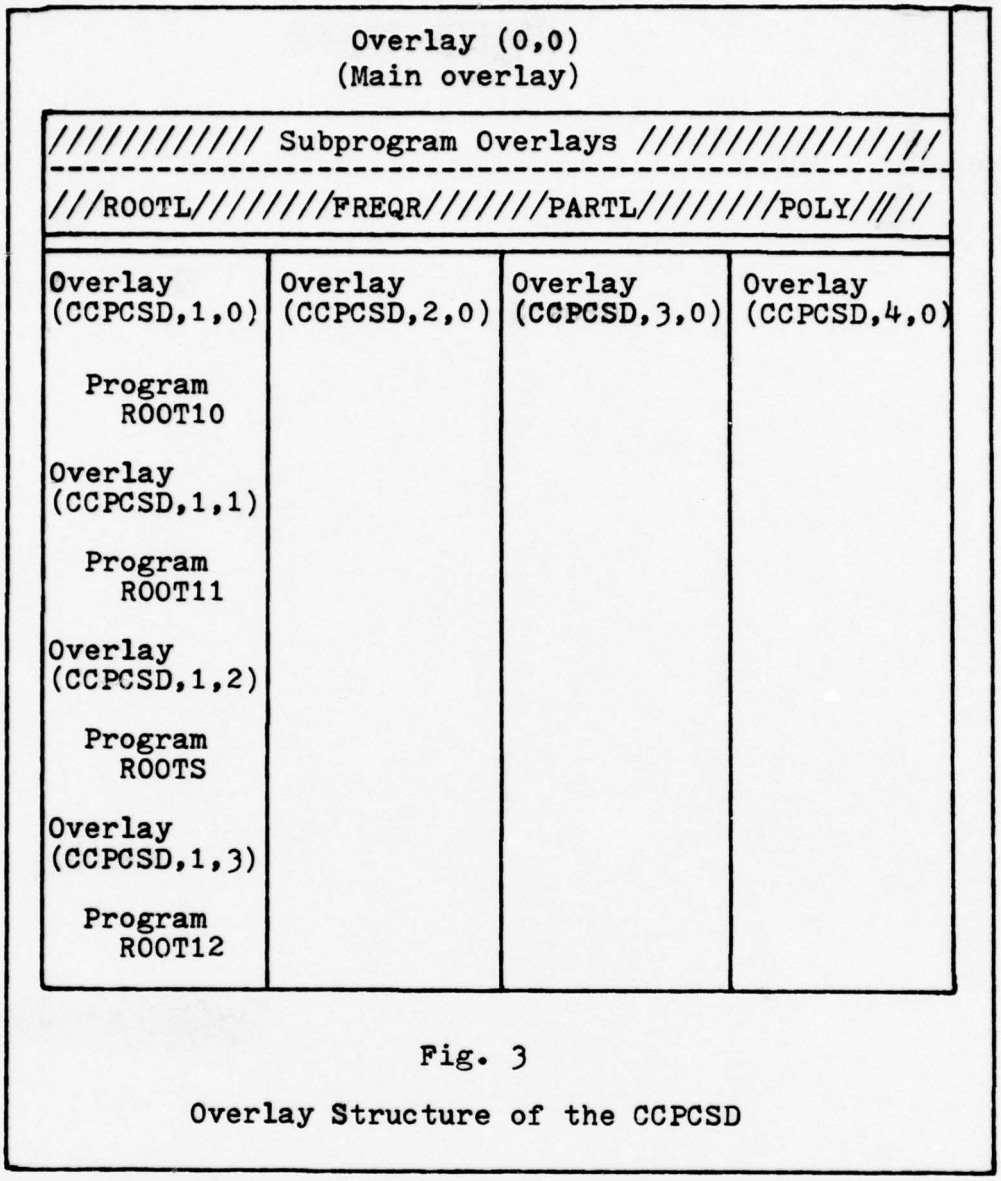


Fig. 3

Overlay Structure of the CCPCSD

Executive (Main) Overlay (CCPCSD,0,0). The executive overlay remains in core at all times, and controls the overall flow of the program. The program statement for Overlay (CCPCSD,0,0) specifies the file names that are used throughout the CCPCSD; i.e., INPUT=100B, OUTPUT=100B, ANSWER, PPLOT, PLOT, CFILE, TAPE5=INPUT, TAPE6=ANSWER, TAPE7=OUTPUT, TAPE8=PPLOT, and TAPE9=100B. File names do not appear in any other overlay.

And, of course, Overlay(CCPCSD,0,0) precedes all other overlays.

The executive overlay provides the following questions, with subsequent direction consistent with the response of the operator:

a. "PLEASE TYPE 'TTY' IF YOU ARE AT AN INTERCOM TERMINAL."

b. "DO YOU WANT AN EXPLANATION OF THE PROGRAM?"

c. "YOU WILL NEED ROOTS AND/OR FACTORS OF THE NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION. IF YOU DO NOT HAVE THESE ROOTS AND/OR FACTORS, TYPE 'POLY'. OTHERWISE, TYPE 'NO'."

d. "TYPE THE SYMBOLIC NAME OF THE RESPONSE YOU DESIRE. EXAMPLE: PARTL"

e. "IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT SUBPROGRAM."

Typing 'POLY' in c above will alert the executive overlay to call Overlay(CCPCSD,4,0), which is a subprogram to factor any polynomial. Typing either 'ROOTL', 'FREQR', 'PARTL', or 'POLY' in d or e above will alert the executive overlay to call Overlay(CCPCSD,1,0), Overlay(CCPCSD,2,0), Overlay(CCPCSD, 3,0), or Overlay(CCPCSD,4,0), respectively. And, as might be expected, typing 'STOP' in e above will terminate the program. The flow chart in Fig. 4 illustrates the operation of the CCPCSD program as controlled by the executive overlay (Overlay (CCPCSD,0,0)).

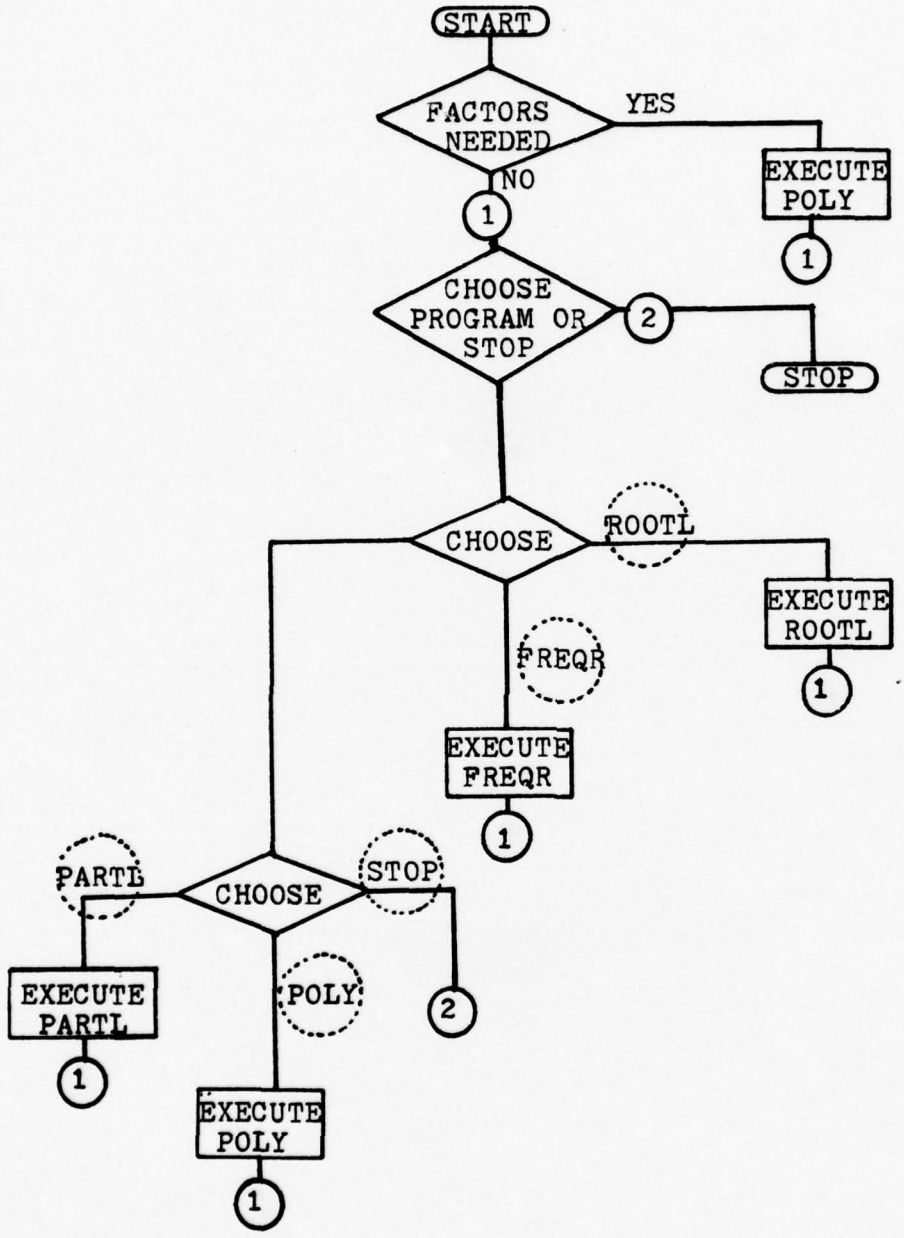


Fig. 4

Flow Chart of the CCPCSD Operation

Subprogram Overlays. The subprogram overlays (Overlay CCPCSD,1,0) - ROOTL; Overlay(CCPCSD,2,0) - FREQR; Overlay (CCPCSD,3,0) - PARTL; and Overlay(CCPCSD,4,0) - POLY) are primary overlays and are called up by the executive overlay (Overlay(CCPCSD,0,0)). They may be considered as totally separable from one another, but, at the same time, not totally independent when considering their individual functions as used in control system design and analysis.

Overlay(CCPCSD,1,0). Overlay(CCPCSD,1,0) represents the subprogram ROOTL, which is a program developed to determine the root locus of a given transfer function by finding the solution to its characteristic equation

$$1 + KG(s)H(s) = 0 \qquad (1)$$

or

$$\frac{K(s-z_1)\ldots(s-z_{m-1})}{(s-p_1)\ldots(s-p_n)} = -1 \qquad (2)$$

as a function of K (K varying from 0 to $\pm$ infinity). Appendix A may be referred to for more specific information about ROOTL's theory and operation.

Structurally, ROOTL consists of the overlays and subroutines shown in Fig. 5. The primary overlay (Overlay (CCPCSD,1,0)) acts as an executive overlay by controlling the overall flow of subprogram ROOTL. The three secondary overlays provide the specific tasks of setting up for a calcomp plot, printing and plotting the root locus, and determining the roots of interest for a specified gain.

Overlay(CCPCSD,1,0) Program ROOT10 Subroutine BANG Subroutine BOX Subroutine GANG1 Subroutine BLEND Subroutine SMULR Subroutine ANG1	Primary Overlay (Directs flow of ROOTL)
Overlay(CCPCSD,1,1) Program ROOT11 Subroutine CNTER Subroutine DNTER Subroutine GROUP	Secondary Overlay (Sets up Calcomp Plot)
Overlay(CCPCSD,1,2) Program ROOTS	Secondary Overlay (Determines Roots of Interest for Specified Gain)
Overlay(CCPCSD,1,3) Program ROOT12 Subroutine BREAK Subroutine SEEK	Secondary Overlay (Trace Root Locus, Print, Plot per Option LL Code)

Fig. 5

Subprogram ROOTL Overlay Structure

Overlay(CCPCSD,2,0). Overlay(CCPCSD,2,0) represents the subprogram FREQR, which will provide a polar plot or Bode plot frequency response for the given transfer functions G(.) and H(.). (The arguments G(.) and H(.) can be either s=jw or z=exp(jwT), where w, omega, is in radians per second and T is the sample period in seconds.) Subprogram FREQR, Overlay (CCPCSD,2,0), consists only of a primary overlay, but it has

six specialized subroutines to complete its task (Fig. 6).

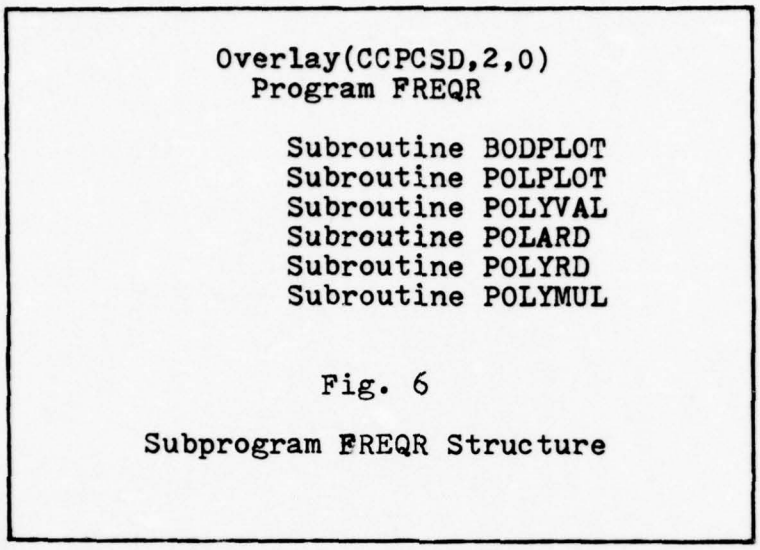


Fig. 6

Subprogram FREQR Structure

Overlay(CCPCSD,3,0). Overlay(CCPCSD,3,0) represents subprogram PARTL, which will calculate the partial fraction expansion coefficients of a transfer function of the form

$$\frac{K(s+a_1+jb_1)(s+a_2+jb_2)\ldots(s+a_m+jb_m)}{(s+g)^L(s+c_1+jd_1)(s+c_2+jd_2)\ldots(s+c_n+jd_n)}, \qquad (3)$$

print the time function of the transfer function, and tabulate or plot the time response. Similar in structure to Overlay (CCPCSD,2,0) for subprogram FREQR, Overlay(CCPCSD,3,0) contains only a primary overlay, along with one subroutine and one FUNCTION routine (Fig. 7).

Overlay(CCPCSD,4,0). Overlay(CCPCSD,4,0) is the last overlay to consider. It represents subprogram POLY, which as its only function, calculates the roots of a polynomial

Overlay(CCPCSD,3,0)
Program PARTL

Subroutine POLAR

FUNCTION FT

Fig. 7

Subprogram PARTL Structure

equation of the form

$$P_n(x) = A_1x^n + A_2x^{n-1} +...+ A_nx + A_{n+1} = 0 \ . \quad (4)$$

It, too, is structured with only a primary overlay, subroutine, and FUNCTION routine (Fig. 8).

Overlay(CCPCSD,4,0)
Program POLY

Subroutine DMULR

FUNCTION RESID

Fig. 8

Subprogram POLY Structure

Standardization of Specific Program Areas

As noted in Chapter I, each of the subprograms (ROOTL, FREQR, PARTL, and POLY) had different input requirements (Table II, pages 13 and 14) and output formats, along with different procedures to begin and stop the programs. The following subsections describes the specific areas of

each program that have been standardized.

Inputs. The inputs for FREQR, PARTL, and POLY are basically the same as the original programmers had written them. Each subprogram consists of printed statements that "ask" the operator to submit the specified input in a required format.

Only the order and coefficients of a desired polynomial are needed as inputs for POLY. PARTL and FREQR basically require the factors of the numerator and denominator of a transfer function, plus specific parameters which control the type, quantity, duration, and limits of the tabulations, printouts, and/or plots. These specific parameters are again "asked for" by the subprograms.

ROOTL, on the other hand, required substantial alteration to allow standardization with the other subprograms. First of all, the front end of Overlay(CCPCSD,1,0) was changed to include a list of options (if the operator desired). The choice of inputting all of the parameters in either one NAMELIST statement, or in a step-by-step manner (the subprogram "asking" for the parameters by associative groupings) was added to assist the operator. Finally, the signs were changed for the parameters XP(.) (real poles), YP(.) (imaginary poles), XZ(.) (real zeros), and YZ(.) (imaginary zeros) to allow ROOTL to accept factors instead of roots.

Therefore, each subprogram has the capabilities of providing a list of options, of "asking" the operator to input specific parameters, and of accepting factors of the

numerator and denominator of the transfer function instead of roots.

Termination. Although the actual method of terminating each of the subprograms seemed small and near insignificant, standardizing the method has been accomplished. For each of the subprograms (ROOTL, FREQR, PARTL, and POLY), the method for termination is to type a '0' (zero) and return the carriage. A note of caution: typing a zero at any point in the program will not necessarily cause termination. The program must present the list of options which include termination before the operator types the zero. However, to terminate the total CCPCSD program (not just the subprogram), the operator may type a '%A'. All prior information and calculated results are lost once the %a is entered and the program stops.

Data Structure. The ideal data structure for this consolidated computer program must be one in which data, such as the numerator coefficients (zeros), denominator coefficients (poles), gain constant, and other similar parameters, are entered one time - all of the subprograms using this one-time input - and that data changes could be made to only the parameters desired to be changed without effecting any of the remaining parameters. However, for reasons stated in the first section of this chapter, the data structure for each of the subprograms was not altered towards achieving a standardized structure. Subprogram ROOTL makes use of the NAMELIST statement for parameters that are entered as inputs, called

out for processing, put back as either the same parameter or as an altered parameter, and used for tabulations and/or plots as required. COMMON statements are used as vehicles for data transfer from overlay to subroutine and overlay to overlay. Subprogram FREQR, PARTL, and POLY made use of COMMON statements only. If a change in a parameter is desired, the subprogram must be re-started and all of the data must be entered; this is unlike ROOTL's NAMELIST data capability of changing only selected parameters (knowing that all others will remain unchanged and at their initialized value).

Standardizing the data structure for the CCPCSD program remains to be a complex task yet to be fully accomplished. Chapter IV discusses some recommendations which affect the data structure selection.

III. Realization, Testing, and Results

Once a total algorithm and flow chart analysis had been formulated, and the total overlay structure for the CCPCSD had been determined, the job of realization of the algorithm as an efficient, usable program became a reality. The following subsections contain a discussion concerning the realization of the total algorithm, the testing criteria and methods, and the results obtained from the testing.

Realization

In order, first, to make sure that continuity throughout all of the subprograms would be accomplished, the program statement for the executive overlay (Overlay(CCPCSD,0,0)) was modified to include all inputs, outputs, and tape files. It reads:

```
PROGRAM CCPCSD(INPUT=100B,OUTPUT=100B,ANSWER,PPLOT,
CFILE,PLOT,TAPE5=INPUT,TAPE6=ANSWER,TAPE7=OUTPUT,
TAPE8=PPLOT,TAPE9=100B)
```

Next, to incorporate the modified file system, the input statements for the subprograms ROOTL and POLY were updated where necessary. Subprograms PARTL and FREQR did not require any modification to their file system. Only the executive overlay (Overlay(CCPCSD,0,0)) contained the program statement with the specified file names required for all of the overlay levels in the CCPCSD program. Hence, each of the separate subprograms were properly corrected for consolidation.

Overlay statements were written; "return" statements were substituted for "stop" and "end" statements (as needed); and

all COMMON and data statements for each of the subprograms were checked for duplication which could cause erroneous results (i.e., storing and manipulating different parameters from the same locations - unintentionally).

The executive overlay (Overlay(CCPCSD,0,0)), subprogram ROOTL, and subprogram POLY were "put together" first. The errors were corrected, and finally these three parts functioned properly together. (Testing is discussed in the following subsection.) Subprograms FREQR and PARTL were added to the consolidated program sequentially, and, after each was added, the same error-correcting test procedure was applied until the total CCPCSD program functioned without errors. At this stage, the CCPCSD program was ready for testing.

Testing

Testing centered basically around four areas:

a. inducing errors intentionally
b. using each option for every subprogram
c. checking for correctiveness of results by comparing with a set of controlled test problems
d. making sure that the termination option functioned properly.

If plots or printed outputs could be "sent" to a central location, (i.e., AFIT computer room, building 640), they were indeed "sent," and were verified as reaching that location by physically picking them up and checking them for correctness.

Executive Overlay (Overlay(CCPCSD,0,0)). Each option in the executive overlay was tested for proper functioning, and each performed properly without fault. Intentional errors, such as typing a symbolic name incorrectly, returning the carriage without any information being previously entered, and omitting part of an instruction, were induced and the response was noted. Error statements were printed out each time an error was entered intentionally, along with a comment for the operator to "try again by typing...". Because the executive overlay must maintain the overall flow of the CCPCSD program, it was extremely important that no error introduced, by mistake or intentionally, would cause the program to jump to another location, or halt without proper cause. The error checks tested operated satisfactory. The executive overlay was not tested for "correctness of results," as it does not perform any calculations, but the terminating operation was performed to insure that the total CCPCSD stopped and that a CP time was given. Hence, the executive overlay (Overlay (CCPCSD,0,0)) was subjected to each of the applicable test areas.

Subprogram Overlays. Initially, each of the subprograms were test for termination upon command. Next, to insure that all of the options for every subprogram worked, and that each worked correctly, the prior-tested examples contained in the original user's guide (ROOTL, FREQR, PARTL,and POLY) were used as the controlled test problems. Every option was tested, and the results of the CCPCSD program were compared with the

results of the sample problems. Finally, intentional errors were introduced randomly throughout each subprogram. Some of the errors used were: exceeding the order of a system beyond its capability; typing in too much data at one time (i.e., too many coefficients, constants, or other parameters); typing an option number that did not exist; and creating a situation in which the computer had "trouble calculating the results within the limits/bounds" (i.e., very small numerator and extremely large denominator of a transfer function with a division process taking place).

Results

The results of testing the CCPCSD program were indeed favorable and encouraging. Each of the subprograms, plus the executive overlay, did terminate upon command. At any point in the program, typing the "%A" did cause complete program termination - as predicted.

All of the options for every subprogram operated as planned; each of the problem results obtained from the CCPCSD program did compare exactly with the results of each of the sample problems; and, finally, the error checking capability of each subprogram proved adequate, but not optimal. Common errors (wrong options, too many constants) were "caught" by a subprogram, and the operator was given instructions to "try again;" however, errors forcing equations to become mathematically unsolvable caused the total CCPCSD program to be terminated.

Although the greater portion of the CCPCSD program worked

well, the subprogram ROOTL does contain one annoying characteristic. The operator is given the choice to enter the data and parameters either as one NAMELIST statement, or as a step-by-step grouping of parameters, with each grouping being "printed out" for the operator to do with as he please. Accomplishing the one-time NAMELIST data is relatively simple if one knows what he wishes to input prior to typing the statement. Accomplishing the step-by-step procedure can be annoying, but its procedure is still simple. As an example, the following statement may be printed out for the operator:

```
"ENTER NUMBER OF LOCUS POINTS COMPUTED,'JK=';
DAMPING RATIO,'ZETA='; AND STEP-SIZE ON RADIAL
SEARCH FOR GIVEN ZETA,'RAD='."
```

If the operator does not wish to change any of the parameters mentioned, he must start in column two, and type '$INPUD $'. Hence, regardless of whether or not data is entered, the operator must start in column two and type '$INPUD', then skip a space and type either another '$' for no data change, or type the data and a '$' at the end of the data. The process is at times annoying, but certainly not unbearable.

All in all, the results from testing the CCPCSD program indicated that the program operated as planned, and that it provided correct results (compared to the testing sample problems). The CCPCSD program is usable.

IV. Conclusions and Recommendations

This chapter provides the reader with the conclusions and recommendations for the CCPCSD program.

Conclusions

Completion of the Consolidated Computer Program for Control System Design (CCPCSD) is definitely a positive step towards providing the control engineer and/or engineering student with the optimum computer-aided design tool. Although the CCPCSD program is not the "optimal program," it *is* definitely a working program, and it *can* be used in its present state for field use in control system design and analysis work.

The overlay structure insures that the core used at any time is at an absolute minimum. The overlay structure also demonstrates current state of the art practices for developing large program/subprogram combinations, and it possesses the capability to add other programs/subprograms as desired with little modification. In short, the overlay structure is effective.

In the view of being a useful product, the CCPCSD does meet the following criteria:

a. It works effectively, and it has understandable instructions.

b. It provides correct results (calculations, plots, tabulations).

c. It contains error diagnosis/statements.

d. It is usable on computers with Fortran IV (Extended) language (CDC 6600, in particular).

The CCPCSD, however, does not meet the criteria for a single data structure that (1) enables the operator to enter the input data one time - this input satisfying all of the subprograms input requirements; and (2) enables information to be passed from one subprogram to another.

The CCPCSD offers considerable advantage (time and convenience) to the engineer or student by providing each design tool (root locus, frequency response, and time response) at his fingertips (one call up of the computer yields all of the programs). And, based upon the results of this thesis (relating to the criteria and tests), the CCPCSD is a useful program for control system design and analysis. It must now be "tried and proven" by serious engineers and/or students - improving its performance wherever possible.

Recommendations

Two distinct areas for immediate/near-future development are recommended:

1. To continue with the present CCPCSD, improving it and expanding its capabilities.
2. To convert the present CCPCSD into an interactive graphics package.

Case 1. If case one is selected, the data structure should be attacked with a goal to make one data structure which is used by all of the programs in the CCPCSD. Provisions should be made to accept added programs without any affect on

the data structure - a challenging job. Also, the CCPCSD should be expanded to accomplish both continuous and digital control system design and analysis. The main overlay should contain the necessary logic, with user-instructions for the added capability. The CCPCSD should also be capable of being run as BATCH jobs.

More programs, as listed in Chapter I, should be added to the CCPCSD, as long as duplications are removed and non-productive programs are eliminated. Prospective avenues for improving and expanding the present CCPCSD are almost limitless.

Case 2. If case 2 is selected, the task of converting the CCPCSD to operate from a graphics terminal would involve restructuring both the data structure and some of the CCPCSD architecture. However, there is a graphics package already developed for program ROOTL (called GROOTL; available in the AFIT library of the CDC 6600 computer system). This graphics package could be a good starting point for a CCPCSD graphics package. Regardless, a graphics package with even the present CCPCSD capabilities is definitely an option urgently needed by engineers and/or engineering students.

Long Range Goals. Long range goals could call for the CCPCSD to be capable, first of all, of operating BATCH, at an intercom terminal, or from a graphics terminal/console. As mentioned before, the amount of inputs required should be at an absolute minimum, and they should be entered only one time to accomplish each task for all of the programs.

Further, the CCPCSD could be expanded to do several tasks

both within and without the realm of control system design and analysis. The CCPCSD could be expanded to do:

a. synthesis of control systems
b. block diagram manipulations (build-up and reduction)
c. optimal estimation and control design (Kalman filter design)
d. statistical analysis
e. continuous-to-digital and digital-to-continuous transformations
f. specific filter designs (Butterworth, Chebyshev, Finite Impulse Response)
g. compensation networks - design and analysis

There are already algorithms that have been developed to accomplish some of the tasks mentioned above. But, actually, before one accepts the challenge of developing the CCPCSD to accomplish all of the tasks listed, the "ground floor" must be done first---attack the data structure; insure that extra programs <u>can</u> be added without altering the data structure, the main CCPCSD, and, if possible, the program being added; and make the CCPCSD concept totally <u>user-oriented</u> and <u>simple</u> to understand. Now the executive overlay (Overlay(CCPCSD,0,0)) can be shaped with the necessary logic to accomplish each newly added task: step-by-step. (Note: All efforts to expand the capabilities of the CCPCSD should, of course, be coordinated with knowledgeable control systems personnel in both the field and the academic environment.)

An evaluation of the algorithms (programs) already available (by job and effectiveness---excellent, adequate, poor) might seem worthy, but past experience has shown that it is nearly impossible to keep track of these algorithms due to changes in location, task, updating, and growth. The confidence factor becomes short termed. Therefore, to help eliminate any frustrating experiences for some other soul who might "trust" that which is in print as being totally correct and up-to-date, the evaluation of the algorithms is not included.

> ". . . up-to-date information must be updated frequently to insure its validity. . ." (unsigned)

Bibliography

1. Control Data Corporation. Fortran Extended Version 4 Reference Manual (Revision H, Publication Number 60305601). Sunnyvale, California: Control Data Corporation, July 1975.

2. Control Data Corporation. SCOPE Reference Manual Version 3.4.4 (Revision J, Publication Number 60307200). Sunnyvale, California: Control Data Corporation, August 1975.

3. Control Data Corporation. Intercom Reference Manual Models 72, 73, 74 Version 4, 6000 Version 4 (Revision E, Publication Number 60307100). Sunnyvale, California: Control Data Corporation, August 1974.

4. Jensen, R.W. and L.P. McNamee, "On Benchmarking CAD Programs." The UCLA Computer Science Department Quarterly, Research and Education, Vol 4, No. 2: 61-68 (April 1976).

5. Thompson, G.W. and K.R. Young, Utilization of a CRT Display Light Pen in the Design of Feedback Control Systems. NASA-CR-11574. Washington: National Aeronautical and Space Administration, May 1972.

6. Young, K.R. The Design of Automatic Control Systems Using Interactive Computer Graphics. unpublished PhD Dissertation. The University of Texas at Austin, May 1974.

7. AFFDL-TR-74-69. Digital Flight Control Systems for Tactical Fighters. Honeywell, Inc. Wright-Patterson Air Force Base, Ohio: Air Force Flight Dynamics Laboratory, July 1974.

8. AFIT/EN. ROOTL User's Manual. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1975.

9. AFIT/EN. PARTL. Heaviside Partial Fraction Expansion and Time Response Program. (Revised) Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, December 1974.

10. AFIT/EN. GROOTL User's Manual. AFIT. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1975.

11. AFIT/ENC. POLY. (Revised User's Manual). AFIT. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, October 1974.

12. AFIT/EN. FREQR. Frequency Response Program. AFIT. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, November 1975.

13. D'Azzo, J. J., and C. H. Houpis. Linear Control System Analysis and Design: Conventional and Modern. McGraw-Hill Book Co., New York, 1975.

APPENDIX A

BRIEF DESCRIPTION OF PROGRAMS ROOTL, FREQR, PARTL, AND POLY (DECEMBER 1976)

Brief Description of Programs ROOTL, FREQR, PARTL, and POLY

This appendix is indeed a brief description of each of the programs - ROOTL, FREQR, PARTL, and POLY. Initially, the approach to this appendix was to obtain every bit of information about each of the programs, and then to write a brief, but thorough, description of each program using the information that had been gathered. However, it is determined that the original user's guides for each of the programs is an "authority" for not only its description, but also for its operation. Hence, with an intent not to draw from that which is already in the original user's guides (duplication of effort is both unprofitable and unnecessary), but yet to provide a better understanding of the programs in their present state (in the CCPCSD), it is decided instead to make the reader aware of each user's guide, and to provide all of the changes to each program since incorporating each into the CCPCSD.

ROOTL (Root Locus Program)

The user's guide for program ROOTL is:

NAME: ROOTL USER'S MANUAL
A DIGITAL COMPUTER PROGRAM TO CALCULATE
AND PLOT THE ROOT LOCUS

DATE: March 1975

DEPARTMENT: Air Force Institute of Technology (AFIT/EN)
Wright-Patterson Air Force Base, OHio 45433

The following changes have been made to ROOTL:

1. Factors instead of roots are now entered for poles and zeros (i.e., for poles (s+3)(s+6): BEFORE - XP(1)=-3,-6 (as roots); NOW - XP(1)=3,6 (as factors).
2. The operator is provided with a list of options (if he desires).
3. The operator may elect to enter the input parameters either in one NAMELIST statement, or in groups of parameters in the form of a step-by-step procedure - several NAMELIST entries (each "asked" by the computer). For each entry, a '$INPUD' must start in column two, and a '$' must be added at the end of the entry.

EXAMPLE: a. Regular one-NAMELIST entry:

```
$INPUD AA=12,BB=13,CC=-10,DD=-14,
N=2,M=1,XP(1)=2,4,YP(1)=0,XZ(1)=3,
YZ(1)=0$
```

b. Reply after being "asked" about option by the step-by-step procedure:

```
$INPUD LL=1$
```

c. Wanting no change to parameters being "asked" by the step-by-step procedure:

```
$INPUD $
```

FREQR (Frequency Response Program)

The user's guide for program FREQR is:

NAME: FREQR
FREQUENCY RESPONSE PROGRAM

DATE: November 1975

DEPARTMENT: Air Force Institute of Technology (AFIT/EN)
Wright-Patterson Air Force Base, Ohio 45433

Program FREQR remains relatively unchanged with the only addition being a list of options (operator requested).

PARTL (Time Response Program)

The user's guide for program PARTL IS:

NAME: PARTL
HEAVISIDE PARTIAL FRACTION EXPANSION AND
TIME RESPONSE PROGRAM

DATE: December 1974 (Revised)

DEPARTMENT: Air Force Institute of Technology (AFIT/EN)
Wright-Patterson Air Force Base, Ohio 45433

The version of program PARTL that is presently in the CCPCSD is, unfortunately, not the latest AFIT version. However, the only change that the updated version has incorporated in it is in the format of the coefficients for the partial fraction expansion. (The version in the CCPCSD was received from the AFIT CDC 6600 computer library as the "latest" version of PARTL.)

Present (CCPCSD) PARTL: (Coefficient printout)

ROOT		POWER	COEFFICIENT	
REAL	IMAG		REAL	IMAG
...	...		...	...
REAL	IMAG		REAL	IMAG
...	...		...	...

Updated version of PARTL: (Coefficient Printout)

FACTOR		POWER		COEFFICIENT	
REAL	IMAG	REAL	IMAG	MAGNITUDE	ANGLE
...	...	...	...	...	...

It is quite possible that the truly updated version of PARTL is available now.

The only change to program PARTL is the termination procedure. A '0' (zero) is used to terminate normally instead of a '6' (six). Also, the operator must remember to dispose his plots to the calcomp plotter.

POLY (Polynomial Factoring Program)

The user's guide for program POLY is:

NAME: POLY
CDC 6600 PROGRAM TO FIND THE ZEROS OF
A POLYNOMIAL

DATE: October 1974 (Revised)

DEPARTMENT: Air Force Institute of Technology (AFIT/ENC)
Wright-Patterson Air Force Base, Ohio 45433

Program POLY does not have any changes or additions since its incorporation into the CCPCSD program.

It should be noted that, even without the program

descriptions or the original user's guide for each of the programs, programs POLY, PARTL, and FREQR (in the CCPCSD) could be used satisfactorily by the operator. ROOTL is more complicated and, because of its input requirements, requires the user's guide as a reference for the symbols (LL, GTOL, GANE, XP(1), and others). These input symbols may be foreign to the operator. Therefore, the user's guides for programs ROOTL, FREQR, PARTL, and POLY should be obtained for references before using the CCPCSD.

APPENDIX B

USER'S MANUAL FOR THE CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPCSD)

(DECEMBER 1976)

Contents

CCPCSD

USER'S MANUAL

Introduction

The CCPCSD is developed to be completely user-oriented. With this in mind, an engineer or student need only know how to call it up (if it is already in the CDC 6600 computer library), and what commands to give the computer to begin executing the program. Therefore, it is assumed that the CCPCSD is in the permanent files in binary deck form and source deck form. A discussion follows now for using both forms. (Underlined commands and instructions indicate the computer commands; the replies from the operator are not underlined.)

CCPCSD In Binary Deck Form

When the CCPCSD is in the CDC 6600 computer in binary form, the commands to use the program are as follows:

(NOTE: PROJX is the file name for the CCPCSD.)

PLEASE LOGIN

LOGIN,(your problem number),(your password),(terminal ID number)

COMMAND - ATTACH,PROJX,(file name PROJX cataloged under),CY=(cycle PROJX cataloged),ID=(ID number PROJX cataloged under if different from yours),PW=(password if used when PROJX cataloged)

COMMAND - ATTACH,CCAUX,ID=X654321,SN=AFIT

COMMAND - LIBRARY,CCAUX

COMMAND - PROJX

The CCPCSD program now begins to execute. From this point, the operator need only follow the directions as they are presented, as they are completely self explanatory.

CCPCSD In Source Deck Form

If the source deck for the CCPCSD is in the permanent files, the following procedure is to be used to make the CCPCSD operational: (PROJX is the file name for the CCPCSD.)

PLEASE LOGIN

LOGIN,(your problem number),(your password),
(terminal ID number)

COMMAND - ATTACH,PROJX,(file name PROJX cataloged under), CY=(cycle PROJX cataloged), ID=(ID number PROJX cataloged under if different from yours),PW=(password if used when PROJX cataloged)

COMMAND - ATTACH,CCAUX,ID=X654321,SN=AFIT

COMMAND - LIBRARY,CCAUX

COMMAND - RU,F=PROJX,FTN

The CCPCSD is now ready for execution. It is compiled (approximately 19 seconds of compilation time), and after being compiled, it begins execution by presenting a program title banner and the statement, "PLEASE TYPE 'TTY' IF YOU ARE AT AN INTERCOM TERMINAL." Again, the operator need only follow the directions as they are presented, as they are completely self explanatory.

Statements The CCPCSD Presents To The Operator

The executive overlay (Overlay(CCPCSD,0,0)) of the

CCPCSD presents the following directions, questions, statements, and explanations in the order given below:

a. "DO YOU WANT AN EXPLANATION OF THE PROGRAM? TYPE 'YES' OR 'NO'."

b. If your reply is 'YES', the following explanation is presented:

" YOU ARE ABOUT TO BEGIN A CONSOLIDATED COMPUTER PROGRAM WHICH CAN BE USED AS AN AID IN DESIGNING AND ANALYZING CONTROL SYSTEMS. YOU MUST HAVE YOUR SYSTEM REDUCED TO A SINGLE TRANSFER FUNCTION OF THE FORM:

$$\frac{K\ (S + Z_1)\ \ldots\ (S + Z_M)}{(S + P_1)\ \ldots\ (S + P_N)}\ ,$$

WHERE K=GAIN CONSTANT, Z=ZEROS, P=POLES OF THE SYSTEM; OR OF THE FORM:

$$\frac{K\ (A_1X^M + A_2X^{M-1} + \ldots + A_MX + A_{M+1}}{(B_1X^N + B_2X^{N-1} + \ldots + B_NX + B_{N+1}}\ .$$

IF YOU HAVE THE LAST FORM SHOWN, YOU WILL BE GIVEN AN OPPORTUNITY TO USE PROGRAM 'POLY' TO OBTAIN FACTORS (ROOTS) OF THE NUMERATOR AND DENOMINATOR FOR YOUR TRANSFER FUNCTION. THEN YOU WILL BE PROVIDED WITH A LIST OF THE INDIVIDUAL SUBPROGRAMS THAT ARE PRESENTLY AVAILABLE FOR YOUR USE. THE INPUT REQUIREMENTS WILL BE SPECIFIED AT THE BEGINNING OF YOUR RUN. AT THE END OF EACH RUN, YOU WILL BE GIVEN THE CHOICE TO STOP, CONTINUE WITH THE PRESENT RUN USING NEW DATA, OR GO TO ANOTHER SUBPROGRAM. IN ANY CASE, YOU WILL BE PROVIDED THE INITIAL LISTING OF SUBPROGRAMS ONLY AT THE BEGINNING; SO, COPY THE LIST AS SOON AS IT IS SHOWN IF YOU INTEND TO DO MORE THAN ONE DESIGN."

c. " YOU WILL NEED THE ROOTS AND/OR FACTORS OF THE NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION. IF YOU DO NOT HAVE THESE ROOTS AND/OR FACTORS, TYPE 'POLY'. OTHERWISE, TYPE 'NO'. (POLY WILL BE AVAILABLE AGAIN AS A SUBPROGRAM.)"

d. " THE FOLLOWING SUBPROGRAMS WITH EXPLAINED OUTPUTS ARE AVAILABLE:

PARTL...PARTIAL FRACTION EXPANSION AND TIME
RESPONSE OUTPUT
FREQR...FREQUENCY RESPONSE WITH POLAR PLOT OR
BODE PLOT OR BOTH
ROOTL...ROOT LOCUS WITH PRINTS OR PLOTS OR BOTH
POLY....DETERMINES ROOTS OF POLYNOMIAL

TYPE THE SYMBOLIC NAME OF THE RESPONSE YOU DESIRE.
EXAMPLE: PARTL."

e. Once you have finished using any one of the programs shown in d above, the following statement is presented:

" IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT SUBPROGRAM."

Each of the statements shown above is easy to understand, and the operator should not have any problem directing the CCPCSD with these statements.

Do's And Don't's

Do: a. Make sure the CCPCSD is in the permanent files before attempting to attach the program. Know if the CCPCSD is in source or binary form.

b. Remember to attach the auxiliary calcomp routines (CCAUX) and to LIBRARY,CCAUX.

c. After commanding the CCPCSD to stop, always check FILES to make sure the file PLOT is disposed. If it is still present, type DISPOSE,PLOT,PT=IBB.

Don't:

a. Do not make any mistake while entering the input data into subprogram ROOTL. This will cause the total CCPCSD program to terminate.

b. Do not forget to double check each entry you make before returning the carriage.

Sample Problem

The following sample problem demonstrates the usefulness and versatility of the CCPCSD.

The system to be analyzed is assumed to be described by the equation

$$C(s) = \frac{4}{s(s^2 + s + 1)(s^2 + 0.4s + 4)}$$

$$= \frac{4}{s(s + 0.5 \pm j0.866)(s + 0.2 \pm j1.99)} \quad .$$

File ANSWER is connected to obtain the printed output of the partial fraction expansion from subprogram PARTL. Options 1 (tabular listing) and 4 (calcomp plot) are chosen from subprogram PARTL. Option 1 (begin new problem), -1 (open-loop response, tabular listing), and -3 (calcomp plot of Bode diagram of the frequency responce) are chosen from subprogram FREQR. Option 3 (root locus, calcomp plot) is chosen for subprogram ROOTL. For the executive overlay, the explanation of the CCPCSD is printed out. For each of the subprograms, the list of options is printed out. The plot file (PLOT) is <u>not</u> disposed at the end of the total run, because subprogram ROOTL automatically disposes the file. (If subprogram ROOTL is <u>not</u> used, then the operator must dispose the plot file by giving the command DISPOSE,PLOT,PT=IBB <u>before</u> LOGOUT.) The terminal printout and calcomp plots are now shown.

```
COMMAND- ATTACH,FRED,OBRIEN,CY=5,PW=FLOB1

COMMAND- ATTACH,CCAUX,ID=X654321,SN=AFIT

 PFN IS
 CCAUX
 PF CYCLE NO. = 001
COMMAND- LIBRARY,CCAUX

COMMAND- CONNECT,ANSWER

COMMAND- FRED
```

PROJX

```
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆START OF CONSOLIDATED COMPUTER PROGRAM◆◆◆◆◆◆◆
◆◆◆◆◆◆◆FOR CONTROL SYSTEM DESIGN.◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

PLEASE TYPE 'TTY' IF YOU ARE AT AN INTERCOM TERMINAL.TTY

  DO YOU WANT AN EXPLANATION OF THE PROGRAM?
 TYPE EITHER 'YES' OR 'NO'.
YES

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
     YOU ARE ABOUT TO BEGIN A CONSOLIDATED COMPUTER
PROGRAM WHICH CAN BE USED AS AN AID IN DESIGNING AND ANALYZING
CONTROL SYSTEMS.  YOU MUST HAVE YOUR SYSTEM REDUCED TO A SINGLE
TRANSFER FUNCTION OF THE FORM◆
```

$$\frac{K\,(S+Z_1)\ldots(S+Z_M)}{(S+P_1)\ldots(S+P_N)}$$

WHERE K = GAIN CONSTANT, Z = ZEROS, P = POLES OF THE SYSTEM,
OR OF THE FORM:

$$\frac{K\,(A_1 X^M + A_2 X^{M-1} + A_3 X^{M-2} + \ldots + A_M X + A_{M+1})}{B_1 X^N + B_2 X^{N-1} + B_3 X^{N-2} + \ldots + B_N X + B_{N+1}}$$

```
    IF YOU HAVE THE LAST FORM SHOWN, YOU WILL BE GIVEN AN
OPPORTUNITY TO USE PROGRAM 'POLY' TO OBTAIN FACTORS (ROOTS)
OF THE NUMERATOR AND DENOMINATOR FOR YOUR TRANSFER FUNCTION.
THEN YOU WILL BE PROVIDED WITH A LIST OF THE INDIVIDUAL SUB-
PROGRAMS THAT ARE PRESENTLY AVAILABLE FOR YOUR USE. THE
INPUT REQUIREMENTS WILL BE SPECIFIED AT THE BEGINNING OF
YOUR RUN. AT THE END OF EACH RUN, YOU WILL BE GIVEN THE
CHOICE TO STOP, CONTINUE WITH THE PRESENT RUN USING NEW
DATA, OR GO TO ANOTHER SUBPROGRAM.  IN ANY CASE, YOU WILL
BE PROVIDED THE INITIAL LISTING OF SUBPROGRAMS ONLY AT THE
BEGINNING; SO, COPY THE LIST AS SOON AS IT IS SHOWN IF YOU
INTEND TO DO MORE THAN ONE DESIGN.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

----------------------------------------------------
       YOU WILL NEED THE ROOTS AND/OR FACTORS OF THE
NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION. IF
YOU DO NOT HAVE THESE ROOTS AND/OR FACTORS, TYPE 'POLY'.
OTHERWISE, TYPE 'NO'. (POLY WILL BE AVAILABLE AGAIN AS A
SUBPROGRAM.) NO

 ----------------------------------------------------

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  THE FOLLOWING SUBPROGRAMS WITH EXPLAINED OUTPUTS ARE
AVAILABLE:

      PARTL...PARTIAL FRACTION EXPANSION WITH TIME RESPONSE OUTPUT.
      FREQR...FREQUENCY RESPONSE WITH POLAR PLOT OR BODE PLOT OUTPUT.
      ROOTL...ROOT LOCUS WITH PRINTS OR PLOTS OR BOTH.
      POLY....DETERMINES ROOTS OF POLYNOMIAL.


   TYPE THE SYMBOLIC NAME OF THE RESPONSE YOU DESIRE.
EXAMPLE:  PARTL PARTL

 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

==========================================================================

++++++++++++++START OF PROGRAM 'PARTL'++++++++++++++
TYPE NO. OF ZEROS, POLES, POWER OF REPEATED ROOT, AND NUMERATOR CONSTANT
0,5,0,4

TYPE REAL AND IMAG PARTS OF DENOMINATOR FACTOR. 0,0

TYPE REAL AND IMAG PARTS OF DENOMINATOR FACTOR. 0.5,0.866

CONJUGATE ASSUMED
TYPE REAL AND IMAG PARTS OF DENOMINATOR FACTOR. 0.2,1.99

CONJUGATE ASSUMED
```

```
             ROOT                        POWER              COEFFICIENT

  REAL              IMAG                               REAL              IMAG
  0.                0.                    1            1.0000            0.
  .50000            .86600                1           -.64516           -.24834
  .50000           -.86600                1           -.64516            .24834
  .20000            1.9900                1            .14515           -.39441E-01
  .20000           -1.9900                1            .14515            .39441E-01

  REAL              IMAG                               MAGNITUDE         ANGLE
  0.                0.                    1            1.0000            0.000
  .50000            .86600                1            .69131           -158.947
  .50000           -.86600                1            .69131            158.947
  .20000            1.9900                1            .15042           -15.201
  .20000           -1.9900                1            .15042            15.201


 THE TIME FUNCTION IS
 F(T)=
           1.0000     EXP( 0.          T)
           1.3826     EXP(-.50000      T) SIN( .86600      *T+ 248.947)
           .30083     EXP(-.20000      T) SIN( 1.9900      *T+ 105.201)


 CALCULATIONS COMPLETE
 FOR TABULAR LISTING TYPE 1
 FOR TELETYPE PLOT TYPE 2
 FOR PRINTER PLOT TYPE 3
 FOR CALCOMP PLOT TYPE 4
 FOR ANOTHER PROBLEM TYPE 5
 TO TERMINATE TYPE 0
 TYPE OPTION1

 TYPE INITIAL TIME, LISTING TIME DURATION, DELTA TIME.0,5,.5

        T                   F(T)
    0.                  -.10730929E-07
    .50000000            .87120957E-02
    1.0000000            .10776403
    1.5000000            .38884255
    2.0000000            .80505999
    2.5000000            1.1809903
    3.0000000            1.3540184
    3.5000000            1.3005925
    4.0000000            1.1378976
    4.5000000            1.0151548
    5.0000000            .99951412
 TYPE OPTION4

 TYPE INITIAL TIME, PLOT DURATION.0,5

 TYPE TITLE - 50 CHARACTERS MAX

SAMPLE CCPCSD---PARTL PORTION
```

```
TYPE NO. OF CHARACTERS, INCL SPACES.29

TYPE OPTION0


        --------END OF PROGRAM 'PARTL'--------


****************************************************************
 IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE
ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT
SUBPROGRAM.
FREQR
 ****************************************************************

================================================================================

**************START OF PROGRAM 'FREQR'**************

 DO YOU WANT A LIST OF OPTIONS AVAILABLE FOR 'FREQR'?
TYPE 'YES' OR 'NO'.
YES


****************************************************************
THE FOLLOWING OPTIONS ARE AVAILABLE FOR PROGRAM 'FREQR':

    1...TO BEGIN A NEW PROBLEM. INPUT OF G(JW) OR G(Z).
   -1...FOLLOWED BY W (OMEGA); CALCULATES G(JW) USING FUNCTION
        ALREADY READ INTO MEMORY AFTER OPTION 1.
   -2...CLOSED-LOOP RESPONSE, C(JW)/R(JW).
    3...PRODUCES TTY PLOT OF OPTIONS -1 OR -2, OR BOTH.
   -3...PRODUCES CALCOMP PLOT OF BODE DIAGRAM (FREQ RESPONSE).
    4...TO READ IN A H(JW), FEEDBACK TRANSFER FUNCTION.
    5...ADDS A TRANSPORT LAG TERM (E(EXP JWT)) TO G(JW).
    6...INPUT NEW FORWARD GAIN CONSTANT ONLY.
   -6...INPUT NEW FEEDBACK GAIN CONSTANT ONLY.
    7...ALLOWS AUTOMATIC DISPOSE BODE PLOTS TO BLDG 640, 'BB'.
    9...TO SEE PRESENT G(S), H(S), OR G(Z) FUNCTIONS.
  -11...SAME AS -1, EXCEPT G(Z) IS CALCULATED.
  -12...SAME AS -2, EXCEPT CLOSED-LOOP Z-DOMAIN IS CALCULATED.
   15...TO SET A SAMPLE PERIOD FOR THE SAMPLED DATA SYSTEM.
    0...CAUSES THE PROGRAM TO STOP.
****************************************************************
TYPE OPTION NUMBER -- 1


  FORWARD TRANSFER FUNCTION -- G(S)

TYPE GAIN CONSTANT K --4
```

```
GAIN CONSTANT =    4.000000

TYPE ORDER OF NUMERATOR -- 0

TYPE NUMBER OF FACTORS OF A GIVEN ORDER -- 1

TYPE ORDER OF THOSE FACTORS -- 0

TYPE  1 COEFFICIENT(S)
1

COEFFICIENT(S) OF NUMERATOR
 1.000000E+00 S( 0)

TYPE ORDER OF DENOMINATOR -- 5

TYPE NUMBER OF FACTORS OF A GIVEN ORDER -- 1

TYPE ORDER OF THOSE FACTORS -- 1

TYPE  2 COEFFICIENT(S)
1,0

COEFFICIENT(S) OF DENOMINATOR
 1.000000E+00 S( 1)  0.            S( 0)

TYPE NUMBER OF FACTORS OF A GIVEN ORDER -- 2

TYPE ORDER OF THOSE FACTORS -- 2

TYPE  6 COEFFICIENT(S)
1,1,1,1,.4,4

COEFFICIENT(S) OF DENOMINATOR
 1.000000E+00 S( 2)  1.000000E+00 S( 1)  1.000000E+00 S( 0)

 1.000000E+00 S( 2)  4.000000E-01 S( 1)  4.000000E+00 S( 0)

TYPE OPTION NUMBER -- -1

TO RESTART PLOT ARRAY STORAGE, ENTER A '1'.

1

TYPE INITIAL, INCREMENTAL, AND FINAL OMEGA.
0.1,.8,8
```

```
    G -- OPEN-LOOP RESPONSE

        W          MAGNITUDE            MAG. IN DB        ANGLE IN DEGREES
        .10        10.074552             20.064515          -96.3423
        .90        1.5051088              3.5513577        -174.5180
       1.70        .71105541             -2.9619311        -259.5217
       2.50        .11175204            -19.034891          -40.5742
       3.30        .16572214E-01        -35.612389          -60.7022
       4.10        .46252095E-02        -46.697372          -68.1662
       4.90        .17258231E-02        -55.260074          -72.3841
       5.70        .76724325E-03        -62.301338          -75.1645
       6.50        .38438311E-03        -68.304714          -77.1566
       7.30        .21018778E-03        -73.547850          -78.6622
TYPE OPTION NUMBER -- -3

 IF YOU WANT A MAG VS W PLOT , ENTER A 0

 IF YOU WANT A DB VS LOG W PLOT,ENTER A 1

 IF YOU WANT BOTH TYPES OF PLOTS, ENTER A 2 2

TYPE IN TITLE OF M-VS-W PLOT, 30 CHARACTERS MAX

TITLE IS SAMPLE CCPCSD---FREQR PORTION

 TYPE IN TITLE OF DB VS LOG W PLOT, 30 CHARACTERS MAX

TITLE IS SAMPLE CCPCSD---FREQR PORTION

TYPE OPTION NUMBER -- 0


      -------END OF PROGRAM 'FREQR'-------


**********************************************************
 IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE
ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT
SUBPROGRAM.
ROOTL

 **********************************************************
```

```
**********START OF PROGRAM 'ROOTL'**********

/**********************************************************************
  DO YOU WANT A LIST OF OPTIONS AVAILABLE FOR 'ROOTL'?
TYPE 'YES' OR 'NO'.YES

**********************************************************
THE FOLLOWING OPTIONS ARE AVAILABLE FOR 'ROOTL':

     0...TO STOP THE PROGRAM
     1...COMPUTES EACH BRANCH OF THE LOCUS OF THE TRANSFER
         FUNCTION OVER SPECIFIED BOUNDED REGION. TABULAR OUTPUT
         FOR BRANCHES, PLUS PLOT, IF DESIRED.
     2...SIMILAR TO '1', EXCEPT STEP SIZE CAN BE SPECIFIED.
     3...SIMILAR TO '1', EXCEPT ZETA, DAMPING RATIO, SPECIFIED.
     4...TO INVESTIGATE PARTS OF LOCUS OF SPECIAL INTEREST.
     5...SIMILAR TO '4', WITH BOUNDARIES OF '1' - '3'; STARTING
         POINT MUST BE SPECIFIED.
     6...TRUNCATED VERSION OF '1'. ONLY ROOTS OF INTEREST FOR A
         SPECIFIED GAIN OF INTEREST (GANE) AND TOLERANCE (GTOL)
         ARE LISTED. NO PLOT GENERATED. GTOL NOT EQUAL TO ZERO.
     7...SIMILAR TO '6'. FOR SPECIFIED DAMPING RATIO (ZETA), ROOTS
         OF INTEREST ARE SPECIFIED. (GANE,GTOL, AND TIME ARE
         NOT SPECIFIED.
**********************************************************

**********************************************************
   NOW YOU WILL HAVE THE CHOICE OF EITHER ENTERING THE
INPUT DATA IN ONE NAMELIST STATEMENT (ASSUMING YOU KNOW EACH
PARAMETER AND HAVE THE LIST PREPARED), OR FOLLOWING A STEP BY
STEP PROCEDURE TO INPUT EACH PARAMETER. IF YOU INTEND TO USE
ONE NAMELIST STATEMENT, TYPE 'ONE'; FOR A STEP BY STEP METHOD,
TYPE 'STEP'.
ONE

----------------------------------------------------------


ENTER NAMELIST INPUD. START IN COLUMN 2 WITH A DOLLAR
SIGN AND INPUD ($INPUD). PUT COMMA BETWEEN EACH ENTRY. END
WITH A DOLLAR SIGN. (EXAMPLE: $INPUD LL=3,N=2,XP(1)=2,4$)
 $INPUD LL=3,N=5,M=0,XP(1)=0,0.5,0.5,0.2,0.2,YP(1)=0,0.866,
 -0.866,1.99,-1.99,GA=4,ZETA=.7,FPN=1,ITITL=29,IPLOT=7,
 MAXDSP=1$

**********************************************************************



ENTER ITITL TITLESAMPLE CCPCSD---ROOTL PORTION


SAMPLE CCPCSD---ROOTL PORTION
```

```
ROOT LOCUS USING OPTION NUMBER  3

 CODE
   0-LOCUS PT.          2-ZERO              4-IMAGINARY AXIS
   1-POLE               3-BREAK PT.         5-SENSITIVITY PT.


 5 POLES AT
   X =  -0.                 Y =   0.
   X =  -.50000             Y =  -.86600
   X =  -.50000             Y =   .86600
   X =  -.20000             Y = -1.9900
   X =  -.20000             Y =  1.9900


 TIME DELAY (TIME) = 0.          , GAIN CONSTANT (GA) =   4.0000000

 DAMPING FACTOR OF INTEREST (ZETA) = .7000

 REGION OF CALCULATION-REAL  -4.50      TO   .500
                       IMAG  -3.50      TO  3.50


                    BRANCH NUMBER   1

 CALCULATION STEP SIZE =  .1000
 PRINTING STEP SIZE    =  .2000

  LOCUS REAL      LOCUS IMAG    DIST TO ORIGIN      GAIN          ZETA     CD
-0.               0.             0.              0.           1.00000    1
-.20000000        0.             .20000000       .166315      1.00000    0
-.40000000        0.             .40000000       .303990      1.00000    0
-.60000000        0.             .60000000       .469664      1.00000    0
-.80000000        0.             .80000000       .725739      1.00000    0
-1.0000000        0.            1.0000000       1.14997       1.00000    0
-1.2000000        0.            1.2000000       1.84509       1.00000    0
-1.4000000        0.            1.4000000       2.94837       1.00000    0
-1.6000000        0.            1.6000000       4.64125       1.00000    0
-1.8000000        0.            1.8000000       7.15894       1.00000    0
-2.0000000        0.            2.0000000      10.8000        1.00000    0
-2.2000000        0.            2.2000000      15.9359        1.00000    0
-2.4000000        0.            2.4000000      23.0208        1.00000    0
-2.6000000        0.            2.6000000      32.6009        1.00000    0
-2.8000000        0.            2.8000000      45.3243        1.00000    0
-3.0000000        0.            3.0000000      61.9501        1.00000    0
-3.2000000        0.            3.2000000      83.3589        1.00000    0
-3.4000000        0.            3.4000000     110.561         1.00000    0
-3.6000000        0.            3.6000000     144.709         1.00000    0
-3.8000000        0.            3.8000000     187.102         1.00000    0
-4.0000000        0.            4.0000000     239.200         1.00000    0
-4.2000000        0.            4.2000000     302.634         1.00000    0
-4.4000000        0.            4.4000000     379.210         1.00000    0
-4.6000000        0.            4.6000000     470.925         1.00000    0
        BOUNDARY
```

BRANCH NUMBER 2

CALCULATION STEP SIZE = .1000
PRINTING STEP SIZE = .2000

LOCUS REAL	LOCUS IMAG	DIST TO ORIGIN	GAIN	ZETA	CD
-.50000000	-.86600000	.99997800	0.	.50001	1
-.30448854	-.82957040	.88368565	.246524	.34457	0
-.11262071	-.88257788	.88973431	.493125	.12658	0
0.	-.94134735	.94134735	.699934	.00000	4
.17382086	-1.0402725	1.0546945	1.15627	.16481	0
.35058902	-1.1338059	1.1867723	1.85652	.29541	0
.53042467	-1.2213151	1.3315258	2.94947	.39836	0

BOUNDARY

BRANCH NUMBER 3

CALCULATION STEP SIZE = .1000
PRINTING STEP SIZE = .2000

LOCUS REAL	LOCUS IMAG	DIST TO ORIGIN	GAIN	ZETA	CD
-.50000000	.86600000	.99997800	0.	.50001	1
-.30448854	.82957040	.88368565	.246524	.34457	0
-.11262071	.88257788	.88973431	.493125	.12658	0
0.	.94134735	.94134735	.699934	.00000	4
.17382086	1.0402725	1.0546945	1.15627	.16481	0
.35058902	1.1338059	1.1867723	1.85652	.29541	0
.53042467	1.2213151	1.3315258	2.94947	.39836	0

BOUNDARY

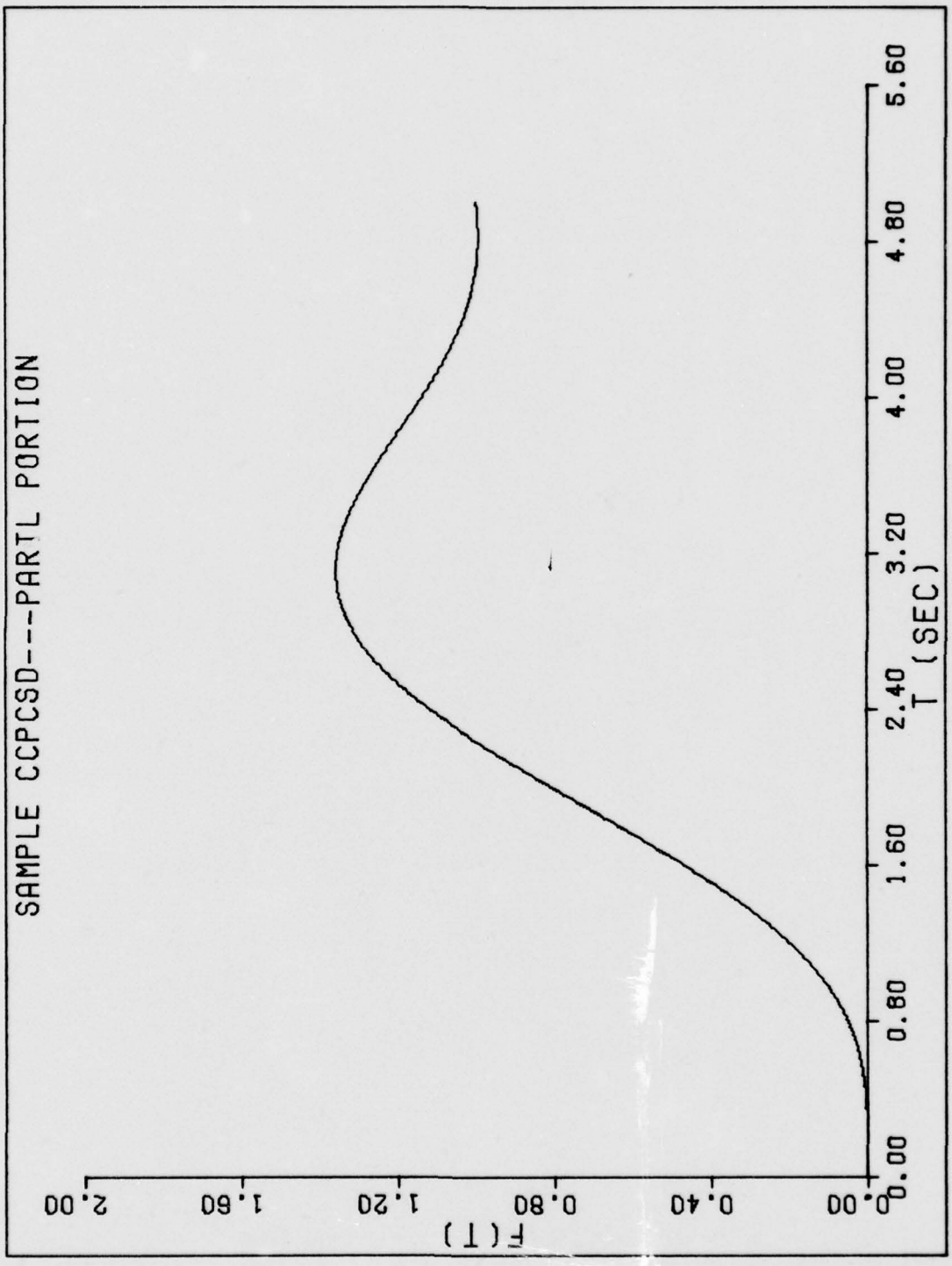
BRANCH NUMBER 4

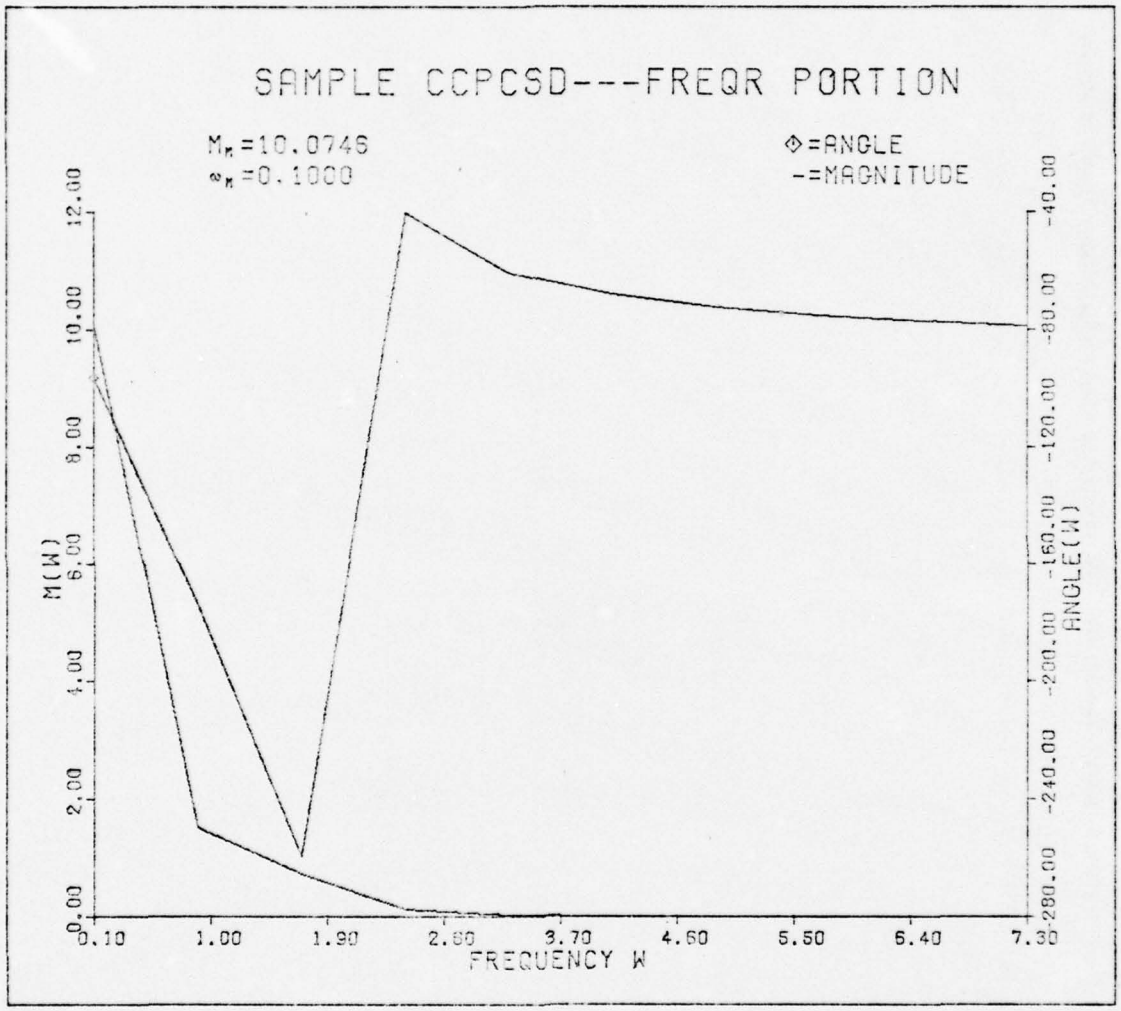
CALCULATION STEP SIZE = .1000
PRINTING STEP SIZE = .2000

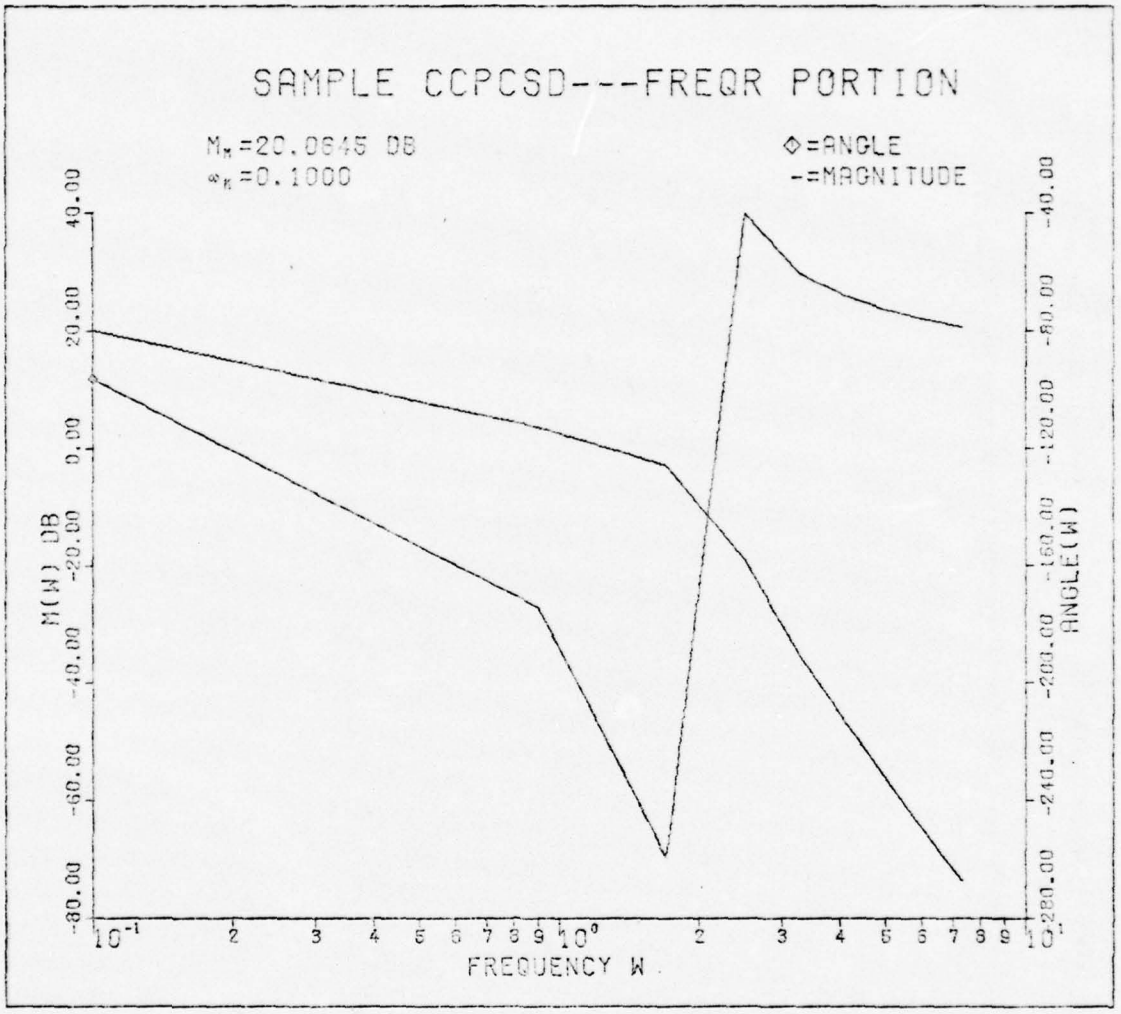
LOCUS REAL	LOCUS IMAG	DIST TO ORIGIN	GAIN	ZETA	CD
-.20000000	-1.9900000	2.0000250	0.	.10000	1
-.39519276	-2.0125619	2.0509955	1.33897	.19268	0
-.55125307	-2.1363737	2.2063482	3.31859	.24985	0
-.67354420	-2.2944421	2.3912604	6.60731	.28167	0
-.77726025	-2.4654006	2.5850210	11.7819	.30068	0
-.87005745	-2.6425529	2.7821009	19.4947	.31273	0
-.95587349	-2.8231994	2.9806289	30.5102	.32070	0
-1.0369288	-3.0060350	3.1798534	45.7200	.32609	0
-1.1145739	-3.1903461	3.3794354	66.1536	.32981	0
-1.1896814	-3.3757066	3.5792090	92.9887	.33239	0
-1.2628424	-3.5618444	3.7790880	127.560	.33417	0

BOUNDARY

```
                    BRANCH NUMBER   5

     CALCULATION STEP SIZE =  .1000
     PRINTING STEP SIZE    =  .2000

  LOCUS REAL      LOCUS IMAG    DIST TO ORIGIN      GAIN          ZETA     CD
-.20000000       1.9900000      2.0000250       0.              .10000    1
-.39519276       2.0125619      2.0509955       1.33897         .19268    0
-.55125307       2.1363737      2.2063482       3.31859         .24985    0
-.67354420       2.2944421      2.3912604       6.60731         .28167    0
-.77726025       2.4654006      2.5850210       11.7819         .30068    0
-.87005745       2.6425529      2.7821009       19.4947         .31273    0
-.95587349       2.8231994      2.9806289       30.5102         .32070    0
-1.0369288       3.0060350      3.1798534       45.7200         .32609    0
-1.1145739       3.1903461      3.3794354       66.1536         .32981    0
-1.1896814       3.3757066      3.5792090       92.9887         .33239    0
-1.2628424       3.5618444      3.7790880       127.560         .33417    0
          BOUNDARY

/**************************************************************
  DO YOU WANT A LIST OF OPTIONS AVAILABLE FOR 'ROOTL'?
TYPE 'YES' OR 'NO'.NO

**************************************************************
   NOW YOU WILL HAVE THE CHOICE OF EITHER ENTERING THE
INPUT DATA IN ONE NAMELIST STATEMENT (ASSUMING YOU KNOW EACH
PARAMETER AND HAVE THE LIST PREPARED), OR FOLLOWING A STEP BY
STEP PROCEDURE TO INPUT EACH PARAMETER. IF YOU INTEND TO USE
ONE NAMELIST STATEMENT, TYPE 'ONE'; FOR A STEP BY STEP METHOD,
TYPE 'STEP'.ONE

--------------------------------------------------------


ENTER NAMELIST INPUD. START IN COLUMN 2 WITH A DOLLAR
SIGN AND INPUD ($INPUD). PUT COMMA BETWEEN EACH ENTRY. END
WITH A DOLLAR SIGN. (EXAMPLE: $INPUD LL=3,N=2,XP(1)=2,4$)
 $INPUD LL=0$

**************************************************************

 1 PLOTS DISPOSED TO TERMINAL 7


      -------END OF PROGRAM 'ROOTL'------


**************************************************************
 IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE
ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT
SUBPROGRAM.
STOP

 **************************************************************
     STOP
```


SAMPLE CCPCSD---PARTL PORTION
F(T)
0.00
0.40
0.80
1.20
1.60
2.00
0.00
0.80
1.60
2.40
3.20
4.00
4.80
5.60
T (SEC)


SAMPLE CCPCSD---FREQR PORTION
M_R=10.0746
ω_R=0.1000
◇=ANGLE
-=MAGNITUDE
M(W)
ANGLE(W)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
-40.00
-80.00
-120.00
-160.00
-200.00
-240.00
-280.00
0.10
1.00
1.90
2.80
3.70
4.60
5.50
6.40
7.30
FREQUENCY W


SAMPLE CCPCSD---FREQR PORTION
M_R=20.0645 DB
ω_R=0.1000
◇=ANGLE
-=MAGNITUDE
M(W) DB
ANGLE(W)
FREQUENCY W
40.00
20.00
0.00
-20.00
-40.00
-60.00
-80.00
-40.00
-80.00
-120.00
-160.00
-200.00
-240.00
-280.00
10^{-1}
10^{0}
10^{1}

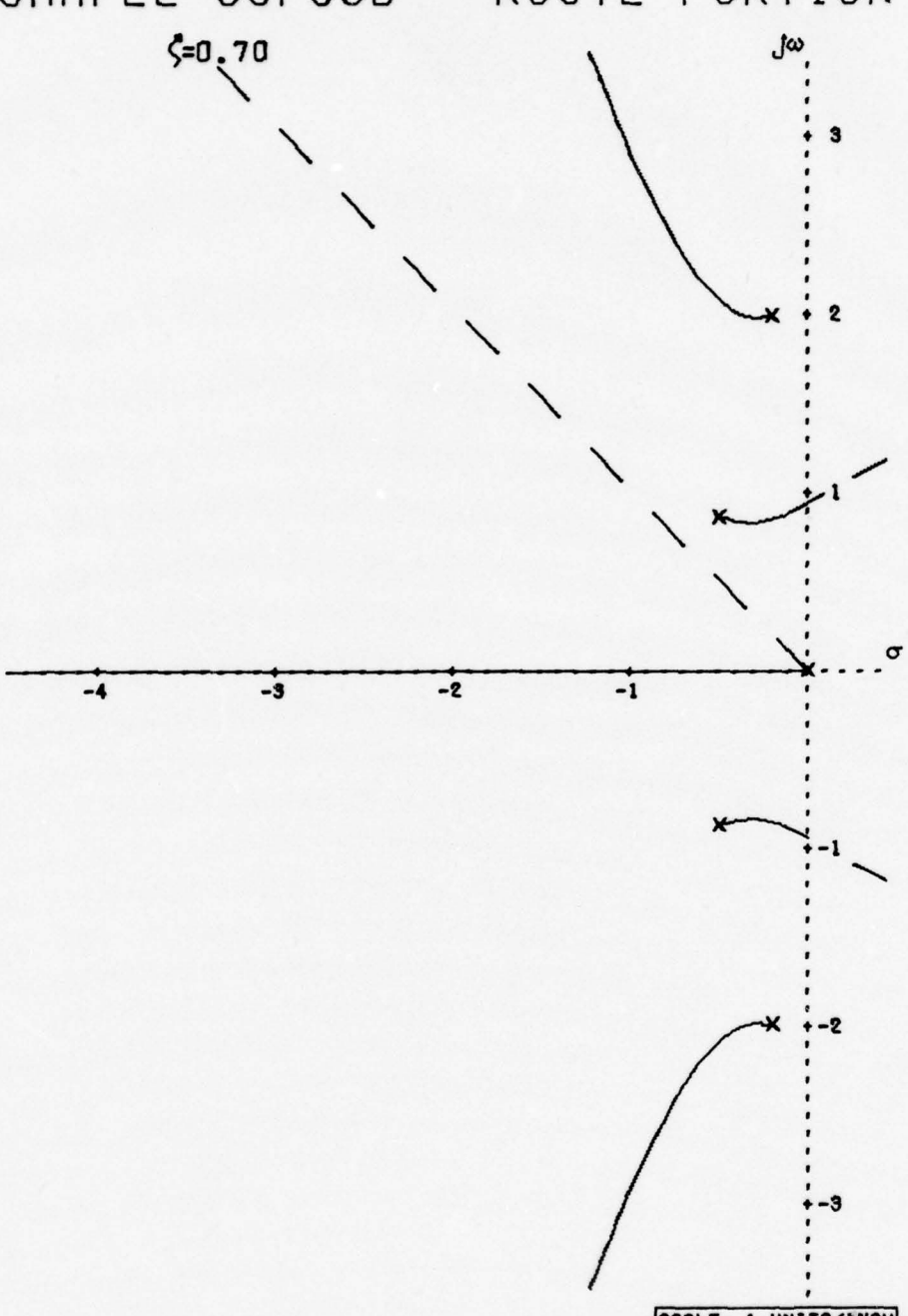


FIGURE 1 - ROOT LOCUS

APPENDIX C

LISTING OF THE CCPCSD

(DECEMBER 1976)

LISTING OF THE CCPCSD

This appendix contains the source listing for the Consolidated Computer Program for Control System Design (CCPCSD). The sequence numbers are on the listing in an effort to maintain the order of the listing, as it is physically long, and, hopefully, to assist anyone who wishes to work with the program.

```
      OVERLAY(CCPCSD,0,0)                                                    000100
      PROGRAM CCPCSD(INPUT=100B,OUTPUT=100B,ANSWER,PPLOT,CFILE,PLOT,         000110
     +TAPE5=INPUT,TAPE7=OUTPUT,TAPE6=ANSWER,TAPE8=PPLOT,TAPE9=100B)          000120
      LOGICAL TTY                                                            000130
      TTY=.TRUE.                                                             000140
      PRINT 888                                                              000150
  888 FORMAT(" ************************************************************ 000160
     +"/" *******START OF CONSOLIDATED COMPUTER PROGRAM*******"/             000170
     +" *******FOR CONTROL SYSTEM DESIGN.*********************"/            000180
     +" ************************************************************ 000190
     +"//)                                                                   000200
      PRINT 999                                                              000210
  999 FORMAT (" PLEASE TYPE 'TTY' IF YOU ARE AT AN INTERCOM TERMINAL.")      000220
      READ 1000,I                                                            000230
 1000 FORMAT (A10)                                                           000240
      IF (I.EQ.3HTTY) GO TO 1                                                000250
    1 PRINT 1013                                                             000260
 1013 FORMAT("   DO YOU WANT AN EXPLANATION OF THE PROGRAM?"/               000270
     +"  TYPE EITHER 'YES' OR 'NO'."/)                                       000280
   15 READ 1014,K                                                            000290
 1014 FORMAT(A10)                                                            000300
      IF(K.EQ.3HYES) GO TO 15                                                000310
      IF(K.EQ.2HNO) GO TO 13                                                 000320
      PRINT 1015                                                             000330
 1015 FORMAT("   SORRY, TRY AGAIN...TYPE EITHER 'YES' OR 'NO'.")            000340
      GO TO 16                                                               000350
   15 PRINT 1001                                                             000360
 1001 FORMAT(" ************************************************************ 000370
     +"/"      YOU ARE ABOUT TO BEGIN A CONSOLIDATED COMPUTER"/              000380
     +" PROGRAM WHICH CAN BE USED AS AN AID IN DESIGNING AND ANALYZING"      000390
```

```
     +/" CONTROL SYSTEMS.  YOU MUST HAVE YOUR SYSTEM REDUCED TO A SINGLE000400
     +"/" TRANSFER FUNCTION OF THE FORM*                                 000410
     +"/"                                                          "     000420
     +/"                  < (S + Z ) ... (S + Z )                  "     000430
     +/"                      1          M                        "     000440
     +/"                -------------------------                  "     000450
     +/"                                                         "     000460
     +/"                   (S + P ) ... (S + P )                 "     000470
     +/"                       1          N          ")                000480
      PRINT 5005                                                        000482
5005 FORMAT(/"  "/                                                     000483
     +/"                                                         "     000490
     +/" WHERE < = GAIN CONSTANT, Z = ZEROS, P = POLES OF THE SYSTEM,"/ 000500
     +" OR OF THE FORM:                                           "/   000510
     +/"                    M      M-1      M-2                     "/   000520
     +"            K (A X + A X    + A X    +...+ A X + A          "/   000530
     +"                1     2        3           M     M+1        "/   000540
     +"           ----------------------------------------------"//     000550
     +"                 N      N-1      N-2                       "/   000560
     +"              B X  + B X    + B X    +...+ B X + B          "/   000570
     +"               1      2        3           N     N+1   "//)     000580
      PRINT 5010                                                        000581
5010 FORMAT(/"    "/                                                   000582
     +"     IF YOU HAVE THE LAST FORM SHOWN, YOU WILL BE GIVEN AN  "/   000590
     +" OPPORTUNITY TO USE PROGRAM 'POLY' TO OBTAIN FACTORS (ROOTS)"/   000600
     +" OF THE NUMERATOR AND DENOMINATOR FOR YOUR TRANSFER FUNCTION."/  000610
     +" THEN YOU WILL BE PROVIDED WITH A LIST OF THE INDIVIDUAL SUB-"/  000620
     +" PROGRAMS THAT ARE PRESENTLY AVAILABLE FOR YOUR USE. THE "/      000630
     +" INPUT REQUIREMENTS WILL BE SPECIFIED AT THE BEGINNING OF"/      000640
     +" YOUR RUN. AT THE END OF EACH RUN, YOU WILL BE GIVEN THE"/       000650
```

```
     +" CHOICE TO STOP, CONTINUE WITH THE PRESENT RUN USING NEW"/        000660
     +" DATA, OR GO TO ANOTHER SUBPROGRAM.  IN ANY CASE, YOU WILL"/      000670
     +" BE PROVIDED THE INITIAL LISTING OF SUBPROGRAMS ONLY AT THE"/     000680
     +" BEGINNING! SO, COPY THE LIST AS SOON AS IT IS SHOWN IF YOU"/     000690
     +" INTEND TO DO MORE THAN ONE DESIGN."/                            000700
     +" ***************************************************")          000710
13    PRINT 1002                                                        000720
1002 FORMAT(//" ---------------------------------------------------     000730
     +"/"         YOU WILL NEED THE ROOTS AND/OR FACTORS OF THE"/       000740
     +" NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION. IF"/        000750
     +" YOU DO NOT HAVE THESE ROOTS AND/OR FACTORS, TYPE 'POLY'."/      000760
     +" OTHERWISE, TYPE 'NO'. (POLY WILL BE AVAILABLE AGAIN AS A "/     000770
     +" SUBPROGRAM.)")                                                  000780
12    READ 1003,TF                                                      000790
1003 FORMAT(A7)                                                        000800
     IF(TF.EQ.4HPOLY) GO TO 3                                          000810
     IF(TF.EQ.2HNO) GO TO 2                                            000820
     PRINT 1010                                                        000830
1010 FORMAT("   SORRY, TRY AGAIN...TYPE 'POLY' OR 'NO'.")              000840
     GO TO 12                                                          000850
3    PRINT *                                                           000860
     PRINT *," ---------------------------------------------"          000870
     CALL OVERLAY(5HCOPOS),4,0,0)                                      000880
     GO TO 3055                                                        000890
2    PRINT *                                                           000900
     PRINT *," ---------------------------------------------"          000910
3055 IF(TTY) PRINT 1005                                                000920
1005 FORMAT(/" ***************************************************"/   000930
     +"   THE FOLLOWING SUBPROGRAMS WITH EXPLAINED OUTPUTS ARE"/       000940
     +" AVAILABLE!"//"       PARTL...PARTIAL FRACTION EXPANSION WITH TIM000950
```

```
     +E RESPONSE OUTPUT."/"        FREQR...FREQUENCY RESPONSE WITH POLAR 000960
     +PLOT OR BODE PLOT OUTPUT."/"        ROOTL...ROOT LOCUS WITH PRINTS 000970
     +OR PLOTS OR BOTH."/"        POLY....DETERMINES ROOTS OF POLYNOMIAL.000980
     +"///"    TYPE THE SYMBOLIC NAME OF THE RESPONSE YOU DESIRE.        000990
     +"/" EXAMPLE:  PARTL")                                              001000
7      READ 1011,J                                                       001010
1011 FORMAT(A10)                                                         001020
     PRINT *," ************************************************"         001030
18    IF(J.EQ.5HPARTL) GO TO 9                                           001040
      IF(J.EQ.5HFREQR) GO TO 11                                          001050
      IF(J.EQ.5HROOTL) GO TO 17                                          001060
      IF(J.EQ.4HPOLY) GO TO 20                                           001070
      IF(J.EQ.4HSTOP) STOP                                               001080
      PRINT 1008                                                         001090
1008 FORMAT("    ERROR...THE SUBPROGRAM YOU HAVE LISTED IS INCORRECT"/   001100
     +" OR NOT AVAILABLE.  RETYPE THE FIVE-LETTER SYMBOLIC NAME"/        001110
     +" FOR THE SUBPROGRAM."/)                                           001120
      GO TO 7                                                            001130
9     CALL OVERLAY(6HCCPCSD,3,0,0)                                       001140
10    PRINT 1009                                                         001150
1009 FORMAT(/" ************************************************"        001160
     +"/"  IF YOU ARE FINISHED, TYPE 'STOP'. IF YOU DESIRE TO USE "/     001170
     +" ANOTHER SUBPROGRAM, TYPE THE CORRECT SYMBOLIC NAME FOR THAT"/    001180
     +" SUBPROGRAM."/)                                                   001190
      READ 1012,J                                                        001200
1012 FORMAT(A10)                                                         001210
     PRINT *," ************************************************"         001220
      IF(J.NE.4HSTOP) GO TO 18                                           001230
      STOP                                                               001240
20    CALL OVERLAY(6HCCPCSD,4,0,0)                                       001250
```

```
      GO TO 10                                                          001260
11    CALL OVERLAY(6HCCPCSD,2,0,0)                                      001270
      GO TO 10                                                          001280
17    CALL OVERLAY(6HCCPCSD,1,0,0)                                      001290
      GO TO 10                                                          001300
      END                                                               001310


      OVERLAY(CCPCSD,1,0)                                               001320
      PROGRAM ROOT10                                                    001330
C                                                                       001340
C     THIS PROGRAM READS INPUT DATA, ESTABLISHES DEFAULT VALUES, AND    001350
C     DIRECTS THE FLOW OF THE OVERALL PROGRAM                           001360
C                                                                       001370
      COMMON /MAIN01/BOUND,DZW(8),FPN,IFOLD,IFPN,ISTAM,ITITL,ITHES,RAD  001380
     +  ,THESI(8),ZETA,MULT                                             001390
      COMMON /MAIN02/CGIN,CNIG,GAG,II,LN,DELPR(50)                      001400
      COMMON /MAIN012/GANE,NOPLOT,GTOL,IIV,LL,V,XAX,YAX,DEL(50),        001410
     +  XDISL(50),XDISR(50),YDISD(50),YDISU(50),XOT,YOT,CB              001420
      COMMON/ANGR2/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME    001430
      COMMON/ANGSS/A,B,ANG,PI                                           001440
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                            001450
      COMMON /ANGJU/ AA,BB,CC,DD,D,DA, JK,SANG,CA                       001460
      LOGICAL NOPLOT                                                    001470
      NAMELIST/INPUD/LL,N,M,GA,TIME,XP,YP,XZ,YZ,DEL,AA,BB,CC,DD,GANE    001480
     1,GTOL,X,Y,LN,JK,SNG,FPN,ITITL,IFOLD,ISTAM,ZETA,ITHES,BOUND,RAD    001490
     2,MULT,IPLOT,MAXOSP,DELPR                                          001500
      DATA GANE/1000./          ,FPN/0/,                                001510
```

```
     1AGIN/6H    GA/,ANIG/6HIN    /,BGIN/6HSENSIT/,BNIG/6HIVITY /        001520
      DATA MULT/0/,AA,BB,CC,DD/.5,3.5,-4.5,-3.5/,PI/3.1415926535/        001530
      DATA E/.0001/,N/1/,M,ITHES/2*0/                                    001540
      DATA NUMPLOT/0/,MAXDSP/4/,FRSTPLT/0./,NNTERM/0/                    001550
      CALL CONNEC(9)                                                     001560
      PRINT 891                                                          001570
  891 FORMAT(//" ================================================================001580
     +================"//" ***************START OF PROGRAM 'ROOTL'*****001590
     +*********")                                                        001600
C                                                                        001610
C     INITIALIZE VALUES                                                  001620
C                                                                        001630
1     GA=1.                                                              001640
      DO 1000 I=1,50                                                     001650
 1000 DELPR(I)=DEL(I)=0                                                  001660
      NTERM=IPLOT=ISTAM=IFOLD=ITITL=LN=GTOL=ZETA=TIME=0                  001670
      LL=J=II=1                                                          001680
      BOUND=SNG=1.                                                       001690
      NOPLOT=.TRUE.                                                      001700
      CGIN=BGIN                                                          001710
      CNIG=BNIG                                                          001720
      JK=50                                                              001730
      RAD=.01                                                            001740
C                                                                        001750
C     READ INPUT DATA                                                    001760
C                                                                        001770
      WRITE(9,3000)                                                      001780
 3004 READ(5,3001) JI                                                    001790
      IF(JI.EQ.3HYES) GO TO 3005                                         001800
      IF(JI.EQ.2HNO) GO TO 3007                                          001810
```

```
      WRITE(9,3003)                                                    001820
C                                                                      001830
C     LIST OF OPTIONS                                                  001840
C                                                                      001850
      GO TO 3004                                                       001860
 3005 WRITE(9,3006)                                                    001870
C                                                                      001880
C     CHOICE OF INPUT FORMATS                                          001890
C                                                                      001900
 3007 WRITE(9,3008)                                                    001910
 3011 READ(5,3009) KL                                                  001920
      WRITE(9,3013)                                                    001930
      IF(KL.EQ.3HONE) GO TO 3012                                       001940
      IF(KL.EQ.4HSTEP) GO TO 3014                                      001950
      WRITE(9,3010)                                                    001960
      GO TO 3011                                                       001970
C                                                                      001980
C     INSTRUCTIONS FOR 'SEPARATE' INPUT. STATEMENTS FOLLOWING.         001990
C                                                                      002000
 3014 WRITE(9,3015)                                                    002010
      WRITE(9,3016)                                                    002020
      READ(5,INPUT)                                                    002030
      WRITE(9,3017)                                                    002040
      READ(5,INPUT)                                                    002050
      WRITE(9,3018)                                                    002060
      READ(5,INPUT)                                                    002070
      WRITE(9,3019)                                                    002080
      READ(5,INPUT)                                                    002090
      WRITE(9,3020)                                                    002100
      READ(5,INPUT)                                                    002110
```

```
      WRITE(9,3021)                                                    002120
      READ(5,INPUT)                                                    002130
      WRITE(9,3022)                                                    002140
      READ(5,INPUT)                                                    002150
      WRITE(9,3023)                                                    002160
      READ(5,INPUT)                                                    002170
      WRITE(9,3024)                                                    002180
      READ(5,INPUT)                                                    002190
      WRITE(9,3025)                                                    002200
      READ(5,INPUT)                                                    002210
      WRITE(9,3026)                                                    002220
      READ(5,INPUT)                                                    002230
      WRITE(9,3027)                                                    002240
      READ(5,INPUT)                                                    002250
      WRITE(9,3028)                                                    002260
      READ(5,INPUT)                                                    002270
      GO TO 3029                                                       002280
C                                                                      002290
C     ONE-TIME INPUT OF NAMELIST DATA                                  002300
C                                                                      002310
 3012 WRITE(9,200)                                                     002320
      READ(5,INPUT)                                                    002330
      WRITE(9,3002)                                                    002340
C                                                                      002350
C     CHANGE SIGNS SO 'FACTORS' ENTERED --- NOT 'ROOTS'                002360
C                                                                      002370
 3029 DO 3040 I=1,M                                                    002380
      XZ(I)=-XZ(I)                                                     002390
      YZ(I)=-YZ(I)                                                     002400
 3040 CONTINUE                                                         002410
```

```
      DO 3041 J=1,N                                                          002420
      XP(J)=-XP(J)                                                           002430
      YP(J)=-YP(J)                                                           002440
 3041 CONTINUE                                                               002450
      IF(EOF(5).NE.0.) LL=0                                                  002460
      IF(LL.GT.4) IPLOT=0                                                    002470
      IF(IPLOT .EQ. 0) GO TO 300                                             002480
      NOPLOT = .FALSE.                                                       002490
      NNTERM=IPLOT                                                           002500
C     IF THIS IS FIRST PLOT TO BE GENERATED,                                 002510
C      INITIALIZE PLOTTER                                                    002520
      IF(FRSTPLT .NE. 0) GO TO 300                                           002530
      FRSTPLT=1.                                                             002540
        CALL PLOT (0.,-3.,-3)                                                002550
        CALL PLOT (0.,1.,-3)                                                 002560
C      DISPOSE PLOTS IF NECESSARY                                            002570
  300 IF(((LL .NE. 0) .AND. ((NUMPLOT .LT. MAXDSP) .OR. (MULT .NE. 0)))      002580
     +  .OR. (NUMPLOT .EQ. 0)) GO TO 301                                     002590
      LLL = 0                                                                002600
      IF(LL.EQ.0) LLL=1                                                      002610
      CALL DSP(NNTERM,LLL)                                                   002620
      WRITE(9,201)NUMPLOT,NNTERM                                             002630
      NUMPLOT=0                                                              002640
  301 IF(LL .EQ. 0) GO TO 1249                                               002650
      TIME=ABS(TIME)                                                         002660
      GA = ABS(GA)                                                           002670
      IF (GA.EQ.1.) GO TO 3                                                  002680
      GA=1./GA                                                               002690
      CGIN=AGIN                                                              002700
      CNIG=ANIG                                                              002710
```

```
C     PAGE EJECT FOR NEW PROBLEM.                                          002720
  3    WRITE(7,174)                                                        002730
C                                                                          002740
C      TEST FOR FIRST TITLE                                                002750
C       0 MEANS NONE                                                       002760
C      >0 MEANS GET ONE                                                    002770
C      <0 MEANS USE THE OLD ONE                                            002780
C                                                                          002790
       IF (ITHES.LE.0) GO TO 4                                             002800
       ITHES=-1                                                            002810
C                                                                          002820
C     READ AND WRITE: TITLES, FIGURE #, OPTION # AND CODE FOR POINTS.      002830
C                                                                          002840
       WRITE(9,202)                                                        002850
       READ (5,181) THESI                                                  002860
  4    IF(ITHES.LT.0)WRITE(7,179)THESI                                     002870
       IF(ITITL .LE. 0) GO TO 5                                            002880
       WRITE(9,203)                                                        002890
       READ(5,181) DZW                                                     002900
       WRITE(7,179)DZW                                                     002910
  5    IFPN=FPN                                                            002920
       IF(IFPN.NE.0)WRITE(7,180)IFPN                                       002930
       WRITE(7,168)LL                                                      002940
       IF (IABS(LL).GE.6) GO TO 400                                        002950
       WRITE(7,1580)                                                       002960
 400   IF (LL.GT.0) GO TO 12                                               002970
C                                                                          002980
C      IF LL IS NEGATIVE XP AND XZ ARE DENOMINATOR AND NUMERATOR           002990
C       COEFFICIENTS OF THE UNFACTORED TRANSFER FUNCTION                   003000
C                                                                          003010
```

```
      LL=IABS(LL)                                                    003020
      GA=GA*ABS(XP(1)/XZ(1))                                         003030
      IF(GA.NE.1.)DGIN=AGIN                                          003040
      IF(GA.NE.1.)DNIG=ANIG                                          003050
      IF (XP(1).EQ.0.) GO TO 1                                       003060
      DO 8 I=1,N                                                     003070
8     XDISR(I+1)=XP(I+1)/XP(1)                                       003080
      XDISR(1)=1.                                                    003090
      CALL SMULR(XDISR,N,XDISL,YDISD)                                003100
      DO 9 I=1,N                                                     003110
      XP(I)=XDISL(I)                                                 003120
9     YP(I)=YDISD(I)                                                 003130
      IF (M.LT.1) GO TO 12                                           003140
      IF (XZ(1).EQ.0.) GO TO 1                                       003150
      DO 10 I=1,M                                                    003160
10    XDISR(I+1)=XZ(I+1)/XZ(1)                                       003170
      XDISR(1)=1.                                                    003180
      CALL SMULR(XDISR,M,XDISL,YDISD)                                003190
      DO 11 I=1,M                                                    003200
      XZ(I)=XDISL(I)                                                 003210
11    YZ(I)=YDISD(I)                                                 003220
C                                                                    003230
C     XP,YP AND XZ,YZ NOW CONTAIN POLES AND ZEROS                    003240
C      (N POLES AND M ZEROS)                                         003250
C                                                                    003260
C    ORDER FACTORS.                                                  003270
C                                                                    003280
12    CALL BLEND                                                     003290
      SIG=1.                                                         003300
      GAG = GA                                                       003310
```

```
      IF(SNG .GE. 0) GO TO 7                                          003320
      GAG = -GAG                                                      003330
      SIG = 0                                                         003340
C                                                                     003350
C     WRITE OUT POLES AND ZEROS TO BE USED                            003360
C                                                                     003370
 7    WRITE(7,189)N                                                   003380
      DO 13 I=1,N                                                     003390
 13   WRITE(7,191)XP(I),YP(I)                                         003400
      IF (M) 16,15,14                                                 003410
 14   WRITE(7,190)M                                                   003420
      DO 15 I=1,M                                                     003430
 15   WRITE(7,191)XZ(I),YZ(I)                                         003440
 16   GAGI=1./GAG                                                     003450
C    WRITE OUT TIME DELAY AND GAIN CONSTANT.                          003460
      WRITE(7,169)TIME,GAGI                                           003470
C                                                                     003480
C     BACK UP ORIGIN IF THIS IS A MULTIPLE PLOT                       003490
C                                                                     003500
      IF (NOPLOT) GO TO 17                                            003510
      IF(MULT .EQ. 0) GO TO 20                                        003520
      CALL PLOT(-4.-XOT,0,-3)                                         003530
      GO TO 17                                                        003540
C                                                                     003550
C     CALL OVERLAY TO DO INITIAL PLOTTING                             003560
C                                                                     003570
 20   NUMPLOT = NUMPLOT + 1                                           003580
 17   CALL OVERLAY(6HCCPCSD,1,1,0)                                    003590
C                                                                     003600
C     THE DEFAULT SIZES ARE                                           003610
```

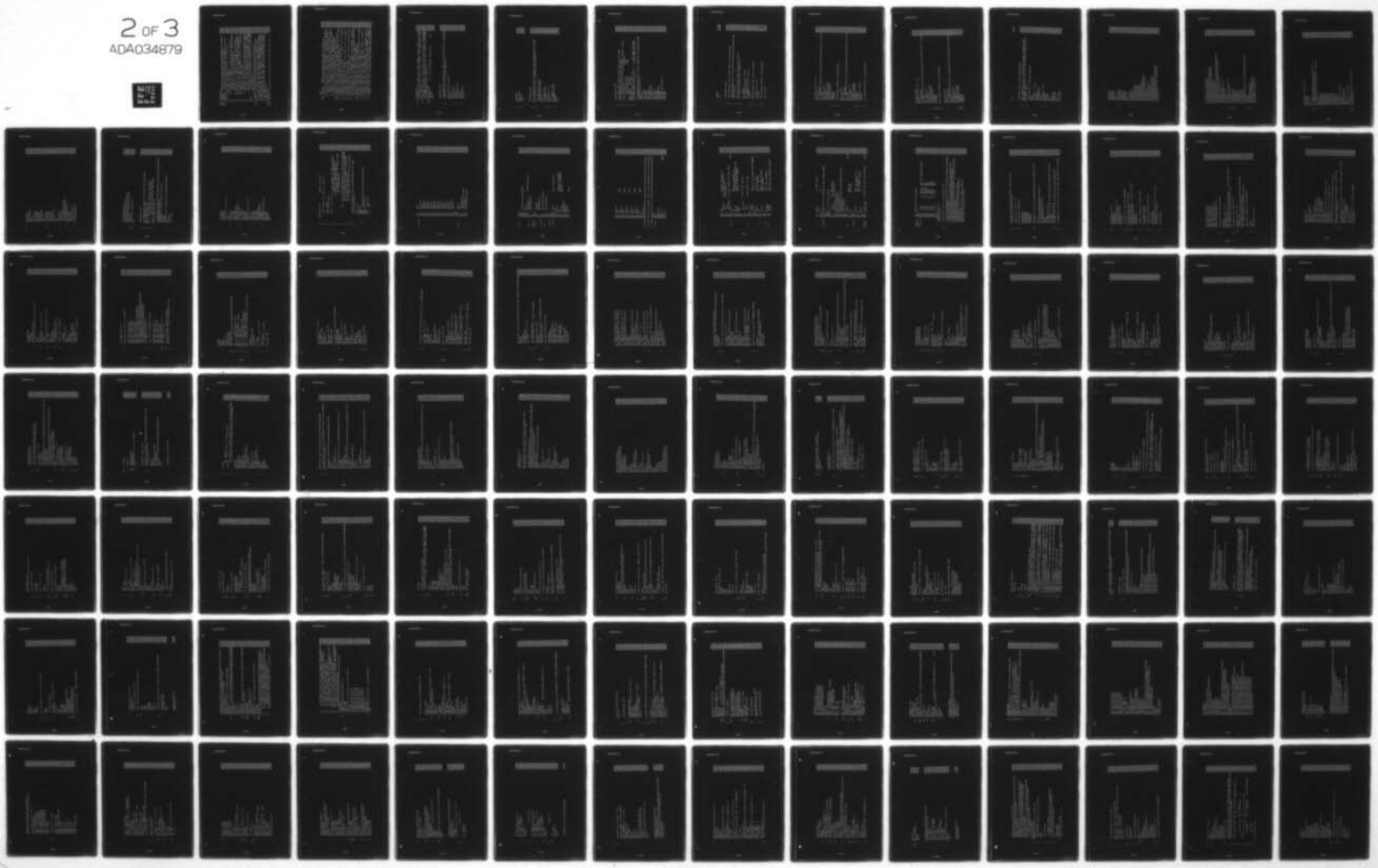
```
C          CALCULATION:  25 POINTS PER INCH OVER THE AREA CONSIDERED     003620
C                        OR .1 WHICH EVER IS LARGER.                     003630
C          PRINTING STEP SIZE:  5 POINTS PER INCH                        003640
C     IN ANY CASE,  THE PRINTING STEP SIZE IS NEVER LESS THAN THE        003650
C     CALCULATION STEP SIZE.                                             003660
C                                                                        003670
      IF(LL .GE. 6) GO TO 22                                             003680
      NTERM = 1                                                          003690
      IF(LL .EQ. 2) NTERM = N+1                                          003700
      DO 1700 I=1,NTERM                                                  003710
      IF(DEL(I) .EQ. 0) DEL(I)=AMAX1( .1,.04/V)                          003720
      IF(DELPR(I) .EQ. 0) DELPR(I) = .2/V                                003730
 1700 IF(DEL(I) .GT. DELPR(I))DELPR(I) = DEL(I)                          003740
      D = DEL(1)                                                         003750
C                                                                        003760
C     IF OPTION 4 OR 5, GO TO OVERLAY TO CALCULATE AND PRINT LOCUS BRANCH003770
C     BUT FOR OPTION 5, FIRST PRINT BOUNDARIES                           003780
C                                                                        003790
      IF(LL-4)22,19,18                                                   003800
C                                                                        003810
C     WRITE OUT THE DESIRED ZETA,GAIN OF INTEREST AND ITS TOLERANCE.     003820
C                                                                        003830
   22 IF(ZETA.NE.0)WRITE(7,177)ZETA                                      003840
      IF(GTOL.NE.(0.).AND.GA.NE.1.)WRITE(7,170)GANE,GTOL                 003850
      Z=GANE/GA                                                          003860
      A=GTOL/GA                                                          003870
      IF(GTOL.NE.0.)WRITE(7,178)Z,A                                      003880
      IF(NOPLOT) GO TO 21                                                003890
C     SET VALUES FOR PLOTTING IN ROOTS                                   003900
      XX=XAX                                                             003910
```

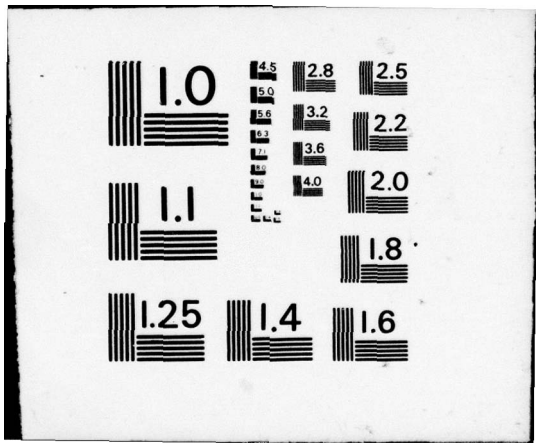
```
      YY=YAX                                                            003920
      G=V                                                               003930
      FX=XOT                                                            003940
      FY=YOT                                                            003950
C                                                                       003960
C     CALL OVERLAY TO CALCULATE THE CLOSED ROOTS AT DESIRED GAIN.       003970
C                                                                       003980
 21    IF((GTOL.NE.0).AND.(TIME.EQ.0))CALL OVERLAY(6HCCPCSD              003990
     + ,1,2,0)                                                          004000
C                                                                       004010
C     IF OPTION #6 OR #7, GO TO A NEW PROBLEM.                          004020
C     BUT FOR OPTION 7, FIRST PRINT BOUNDARIES.                         004030
C                                                                       004040
      IF (LL.EQ.6) GO TO 1                                              004050
C     WRITE OUT REGION OF CALCULATION                                   004060
 18    WRITE(7,172)CC,AA,DD,BB                                          004070
      IF(LL .EQ. 7) GO TO 1                                             004080
C                                                                       004090
C     CALL OVERLAY TO TRACE ROOT LOCUS AND PRINT AND PLOT AS DESIRED    004100
C                                                                       004110
 19    CALL OVERLAY(6HCCPCSD,1,3,0)                                     004120
C                                                                       004130
C     MOVE OFF PLOT AND DUMP FINAL PLOT BUFFER                          004140
C       (ORIGIN WILL HAVE TO BE RESET IF MULTIPLE PLOT WANTED)          004150
C                                                                       004160
      IF (NOPLOT) GO TO 1                                               004170
      CALL PLOT(XOT+4.,0,-3)                                            004180
      GO TO 1                                                           004190
C                                                                       004200
C     FORMATS                                                           004210
```

```
C                                                                       004220
 150  FORMAT (//,5X,31HROOT LOCUS USING OPTION NUMBER ,I2)              004230
 1580 FORMAT (/,6X,4HCODE,/,8X,11H0-LOCUS PT.,10X,6H2-ZERO,15X,15H4-IMAG004240
     1INARY AXIS,/,8X,6H1-POLE,15X,11H3-BREAK PT.,10X,17H5-SENSITIVITY P004250
     2T.)                                                               004260
 159  FORMAT (//,6X,19HTIME DELAY (TIME) =,G10.4,22H, GAIN CONSTANT (GA)004270
     1 =,G16.8)                                                         004280
 170  FORMAT (/,6X,25HGAIN OF INTEREST (GANE) =,G13.4,5X,11H+OR- GTOL =,004290
     1G12.4)                                                            004300
 172  FORMAT (/,6X,27HREGION OF CALCULATION-REAL ,G10.3,2X,3HTO ,G10.3,/004310
     1,28X,5HIMAG ,G10.3,2X,3HTO ,G10.3)                                004320
174   FORMAT (1H1)                                                      004330
 177  FORMAT (/,6X,35HDAMPING FACTOR OF INTEREST (ZETA) =,G10.4)        004340
 178  FORMAT (/,6X,29HSENSITIVITY OF INTEREST (K) =,G13.4,1X,11H+OR- KTO004350
     1L =,G12.4)                                                        004360
 179  FORMAT("0",8A10)                                                  004370
 180  FORMAT (/,5X,6HFIGURE,I4)                                         004380
 181  FORMAT(8A10)                                                      004390
 189  FORMAT (//,4X,I3,1X,8HPOLES AT)                                   004400
 190  FORMAT (//,4X,I3,1X,8HZEROS AT)                                   004410
 191  FORMAT (8X,3HX =,G13.5,9X,3HY =,G13.5)                            004420
 200  FORMAT(" ENTER NAMELIST INPUD. START IN COLUMN 2 WITH A DOLLAR"/  004430
     +" SIGN AND INPUD ($INPUD). PUT COMMA BETWEEN EACH ENTRY. END"/    004440
     +" WITH A DOLLAR SIGN. (EXAMPLE: $INPUD LL=3,N=2,XP(1)=2,4$)"/)    004450
 201  FORMAT(I3," PLOTS DISPOSED TO TERMINAL",I2)                       004460
 202  FORMAT(".ENTER THESI TITLE")                                      004470
 203  FORMAT(".ENTER ITITL TITLE")                                      004480
3000 FORMAT(/"//##################################################    004490
     +"/"   DO YOU WANT A LIST OF OPTIONS AVAILABLE FOR 'ROOTL'?"/      004500
     +" TYPE 'YES' OR 'NO',")                                          004510
```

AD-A034 879 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPC--ETC(U)
DEC 76 F L O'BRIEN
UNCLASSIFIED GE/EE/76D-38 NL

2 OF 3
ADA034879




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

```
3001 FORMAT(A10)                                                        004520
3002 FORMAT(" ##########################################################"/)  004530
3003 FORMAT("    SORRY, TRY AGAIN...TYPE 'YES' OR 'NO'.")                  004540
3006 FORMAT(/" ********************************************************   004550
     +"/" THE FOLLOWING OPTIONS ARE AVAILABLE FOR 'ROOTL':"//             004560
     +"     0...TO STOP THE PROGRAM"/                                     004570
     +"     1...COMPUTES EACH BRANCH OF THE LOCUS OF THE TRANSFER"/       004580
     +"         FUNCTION OVER SPECIFIED BOUNDED REGION. TABULAR OUTPUT"/004590
     +"         FOR BRANCHES, PLUS PLOT, IF DESIRED."/                    004600
     +"     2...SIMILAR TO '1', EXCEPT STEP SIZE CAN BE SPECIFIED."/      004610
     +"     3...SIMILAR TO '1', EXCEPT ZETA, DAMPING RATIO, SPECIFIED."/004620
     +"     4...TO INVESTIGATE PARTS OF LOCUS OF SPECIAL INTEREST."/      004630
     +"     5...SIMILAR TO '4', WITH BOUNDARIES OF '1' - '3'; STARTING"/004640
     +"         POINT MUST BE SPECIFIED."/                                004650
     +"     6...TRUNCATED VERSION OF '1'. ONLY ROOTS OF INTEREST FOR A"/004660
     +"         SPECIFIED GAIN OF INTEREST (GANE) AND TOLERANCE (GTOL)"/004670
     +"         ARE LISTED. NO PLOT GENERATED. GTOL NOT EQUAL TO ZERO."/004680
     +"     7...SIMILAR TO '6'. FOR SPECIFIED DAMPING RATIO (ZETA), ROOT004690
     +S"/"         OF INTEREST ARE SPECIFIED. (GANE,GTOL, AND TIME ARE"/004700
     +"         NOT SPECIFIED."/                                          004710
     +" ********************************************************"/)      004720
3008 FORMAT(" ********************************************************   004730
     +"/"    NOW YOU WILL HAVE THE CHOICE OF EITHER ENTERING THE"/        004740
     +" INPUT DATA IN ONE NAMELIST STATEMENT (ASSUMING YOU KNOW EACH"/    004750
     +" PARAMETER AND HAVE THE LIST PREPARED), OR FOLLOWING A STEP BY"/   004760
     +" STEP PROCEDURE TO INPUT EACH PARAMETER. IF YOU INTEND TO USE"/    004770
     +" ONE NAMELIST STATEMENT, TYPE 'ONE'; FOR A STEP BY STEP METHOD,"/004780
     +" TYPE 'STEP',")                                                   004790
3009 FORMAT(A10)                                                        004800
3013 FORMAT(" ---------------------------------------------------------  004810
```

```
     +"//)                                                                  004820
3010 FORMAT("    SORRY, TRY AGAIN...TYPE 'ONE' OR 'STEP'.")                 004830
3015 FORMAT(/" **********IMPORTANT**********"/                              004840
     +" BE SURE TO START EACH ENTRY IN COLUMN 2. BEGIN WITH '$INPUD';"/     004850
     +" SKIP A SPACE: ENTER THE DATA; END WITH A DOLLAR SIGN, $. "/         004860
     +" EXAMPLE:  $INPUD LL=3,N=2,M=3,XP(1)=2,4$  ALSO, IF YOU DO NOT"/     004870
     +" WANT TO ENTER/CHANGE THE PARTICULAR PARAMETER SHOWN, ENTER"/        004880
     +"  $INPUD $. TO STOP THE PROGRAM, ENTER ' $INPUD LL=0$."/             004890
     +" **********"/)                                                      004900
3016 FORMAT(/" CHOOSE OPTION (0 THRU 7): PRECEDE ENTRY BY 'LL='."/)        004910
3017 FORMAT(/" ENTER NUMBER OF POLES (DENOMINATOR ORDER) AS 'N=', AND"/    004920
     +" NUMBER OF ZEROS (NUMERATOR ORDER) AS 'M='."/)                      004930
3018 FORMAT(/" ENTER DENOMINATOR FACTORS: REAL PARTS AS 'XP( )=',AND"/     004940
     +" IMAGINARY PARTS AS 'YP( )='."/)                                    004950
3019 FORMAT(/" ENTER NUMERATOR FACTORS: REAL PARTS AS 'XZ( )=', AND"/      004960
     +" IMAGINARY PARTS AS 'YZ( )='."/)                                    004970
3020 FORMAT(/" ENTER THE OPEN-LOOP GAIN, 'GA=': SIGN OF OPEN-LOOP GAI,     004980
     +"/" 'SNG=': AND TIME DELAY, 'TIME='."/)                              004990
3021 FORMAT(/" ENTER THE RH BOUNDARY, 'AA=': UPPER BOUNDARY, 'BB=';"/      005000
     +" LH BOUNDARY,'CC=-': AND LOWER BOUNDARY, 'DD=-': EACH SPECIFYING    005010
     +"/" THE BOUNDARY FOR THE REGION OF CALCULATION."/)                   005020
3022 FORMAT(/" ENTER BOUNDARY SCALE FACTOR, 'BOUND='."/)                   005030
3023 FORMAT(/" ENTER REAL AXIS LENGTH, 'IFOLD=': CALCULATION STEP SIZE,    005040
     +"/" 'DEL=': AND PRINT STEP SIZE, 'DELPR='."/)                        005050
3024 FORMAT(/" ENTER CLOSED-LOOP GAIN OF INTEREST, 'GANE=': AND THE"/      005060
     +" REGION OF INTEREST ABOUT GANE, 'GTOL='."/)                         005070
3025 FORMAT(/" ENTER REAL PART STARTING POINT, 'X=': IMAGINARY PART"/      005080
     +" STARTING POINT, 'Y=': AND STARTING POINT TYPE NUMBER, 'LN='."/)     005090
3026 FORMAT(/" ENTER NUMBER OF LOCUS POINTS COMPUTED, 'JK=': DAMPING"/     005100
     +" RATIO, 'ZETA=': AND STEP-SIZE ON RADIAL SEARCH FOR GIVEN ZETA,"/   005110
```

```
     +" 'RAD=',"/)                                                          005120
3027 FORMAT(/" FOR PLOTS: ENTER FIGURE NUMBER, 'FPN=': FIRST TITLE FLAG005130
     +,"/" 'ITHES=': SECOND TITLE FLAG, 'ITITL=': AND THIRD TITLE FLAG,"005140
     +/" 'ISTAM=',"/)                                                       005150
3028 FORMAT(/" FOR PLOTS: ENTER MULTIPLE PLOT FLAG, 'MULT=': PLOTTING"/005160
     +" TERMINAL INDICATOR (7 FOR 3B AT BLDG 640), 'IPLOT=': AND DISPOSE005170
     +"/" LIMIT FOR PLOTS, 'MAXDSP=',"/)                                    005180
1249 WRITE(9,3050)                                                          005190
3050 FORMAT(//"       -------END OF PROGRAM 'ROOTL'------        "//)      005200
     RETURN                                                                 005210
     END                                                                    005220


     SUBROUTINE BANG                                                        005230
C                                                                           005240
C    THIS SUBROUTINE COMPUTES THE ANGLE OF A VECTOR FROM ITS RECTANGULA005250
C    COMPONENTS, A, B, BY FINDING ARCTAN(B/A)                               005260
C                                                                           005270
     COMMON/ANGSS/A,B,ANG,PI                                                005280
     ANG=0.                                                                 005290
     IF(A) 2,1,3                                                            005300
 1   IF(B) 5,6,4                                                            005310
 2   BAA=B/A                                                                005320
     ANG=ATAN(BAA)+PI                                                       005330
     GO TO 6                                                                005340
3    BAA=B/A                                                                005350
     ANG=ATAN(BAA)                                                          005360
     GO TO 6                                                                005370
```

```
4       ANG=PI/2.                                                       005380
        GO TO 6                                                         005390
5       ANG=-PI/2.                                                      005400
6       RETURN                                                          005410
        END                                                             005420


        SUBROUTINE BOX                                                  005430
C                                                                       005440
C     THIS SUBROUTINE REDUCES AN ANGLE TO ITS PRIMARY VALUE BETWEEN     005450
C     PLUS AND MINUS PI (I.E. REMOVES MULTIPLES OF PI)                  005460
C                                                                       005470
        COMMON/ANGSS/A,B,ANG,PI                                         005480
        LM=ABS(ANG)/6.283185                                            005490
        T=LM                                                            005500
        IF (ANG) 1,5,2                                                  005510
1       ANG=ANG+T*2.*PI                                                 005520
        GO TO 3                                                         005530
2       ANG=ANG-T*2.*PI                                                 005540
3       T=1.                                                            005550
        IF (ANG+PI) 1,5,4                                               005560
4       IF (ANG-PI) 5,2,2                                               005570
5       RETURN                                                          005580
        END                                                             005590
```

```
      SUBROUTINE GANG1                                                      005600
C                                                                           005610
C     THIS SUBROUTINE COMPUTES PHASE ANGLE OF TEST POINT                    005620
C      GANG=ANG((T-Z(1))+(T-Z(2))+...+(T-Z(M))) -                           005630
C           ANG((T-P(1))+(T-P(2))+...+(T-P(N))) - PI*SIG - TAU*YT           005640
C     WHERE: T=(XT,YT)                    TEST POINT                        005650
C            Z(J)=(XZ(J),YZ(J))             ZERO                          005660
C            P(I)=(XP(I),YP(I))             POLE                          005670
C     NOTE: SIG=1.    FOR GA>0.                                             005680
C     NOTE: SIG=0.    FOR GA<0.                                             005690
C     AS DEFINED ABOVE, GANG=0 DEG. - - - - LOCUS POINT FOR GA>0.           005700
C     AS DEFINED ABOVE, GANG=180 DEG. - - - LOCUS POINT FOR GA<0.           005710
C                                                                           005720
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME        005730
      COMMON/ANGSS/A,B,ANG,PI                                               005740
      GANG=0.                                                               005750
      DO 2 I=1,N                                                            005760
      A=X-XP(I)                                                             005770
      B=Y-YP(I)                                                             005780
      CALL BANG                                                             005790
      GANG=GANG-ANG                                                         005800
      IF (I-M) 1,1,2                                                        005810
1     A=X-XZ(I)                                                             005820
      B=Y-YZ(I)                                                             005830
      CALL BANG                                                             005840
      GANG=GANG+ANG                                                         005850
2     CONTINUE                                                              005860
      ANG=GANG-PI*SIG-TIME*Y                                                005870
      CALL BOX                                                              005880
      GANG=ANG                                                              005890
```

```
      RETURN                                                                005900
      END                                                                   005910


      SUBROUTINE BLEND                                                      005920
C                                                                           005930
C     THIS SUBROUTINE ORDERS ROOTS PRIOR TO PLOTTING LOCUS                  005940
C                                                                           005950
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME        005960
      DATA E/.0001/                                                         005970
C                                                                           005980
C     IF POLE OR ZERO IS ALMOST REAL, MAKE IT REAL.                         005990
C     SETS IMAGINARY PARTS <10**-4 EQUAL TO 0.                              006000
C                                                                           006010
      DO 1 I=1,N                                                            006020
      IF (ABS(YP(I)).LE.E) YP(I)=0.                                         006030
      IF (M.LT.I) GO TO 1                                                   006040
      IF (ABS(YZ(I)).LE.E) YZ(I)=0.                                         006050
1     CONTINUE                                                              006060
C                                                                           006070
C     ARRANGE COMPLEX POLE PAIRS IN SEQUENCE                                006080
C                                                                           006090
      IF (N.LE.2) GO TO 8                                                   006100
      NNN=N-1                                                               006110
C                                                                           006120
C     GET CONJUGATE PAIRS OF POLES ADJACENT                                 006130
C                                                                           006140
      DO 4 I=1,NNN                                                          006150
```

```
      IF (YP(I).EQ.0.) GO TO 4                                          006160
      MMM=I+1                                                           006170
      DO 3 JM=MMM,N                                                     006180
      IF ((ABS(XP(I)-XP(JM)).GT.E).OR.(ABS(YP(I)+YP(JM)).GT.E)) GO TO 3 006190
      IIM=JM-I                                                          006200
      DO 2 JJM=1,IIM                                                    006210
      IPM=JM-JJM                                                        006220
      XP(IPM+1)=XP(IPM)                                                 006230
2     YP(IPM+1)=YP(IPM)                                                 006240
      YP(I+1)=-YP(I)                                                    006250
3     CONTINUE                                                          006260
4     CONTINUE                                                          006270
C                                                                       006280
C     MOVE ALMOST IDENTICAL POLES ADJACENT                              006290
C                                                                       006300
      DO 7 I=1,NNN                                                      006310
      MMM=I+1                                                           006320
      DO 6 JM=MMM,N                                                     006330
      IF ((ABS(XP(I)-XP(JM)).GT.E).OR.(ABS(YP(I)-YP(JM)).GT.E)) GO TO 6 006340
      IIM=JM-I                                                          006350
      DO 5 JJM=1,IIM                                                    006360
      IPM=JM-JJM                                                        006370
      XP(IPM+1)=XP(IPM)                                                 006380
5     YP(IPM+1)=YP(IPM)                                                 006390
6     CONTINUE                                                          006400
7     CONTINUE                                                          006410
8     IF (M.LE.2) RETURN                                                006420
      NNN=M-1                                                           006430
C                                                                       006440
C     GET CONJUGATE PAIRS OF ZEROS ADJACENT                             006450
```

```
C                                                                       006460
      DO 11 I=1,NNN                                                     006470
      IF (YZ(I).EQ.0.) GO TO 11                                         006480
      MMM=I+1                                                           006490
      DO 10 JM=MMM,M                                                    006500
      IF ((ABS(XZ(I)-XZ(JM)).GT.E).OR.(ABS(YZ(I)+YZ(JM)).GT.E)) GO TO 10006510
      IIM=JM-I                                                          006520
      DO 9 JJM=1,IIM                                                    006530
      IPM=JM-JJM                                                        006540
      XZ(IPM+1)=XZ(IPM)                                                 006550
9     YZ(IPM+1)=YZ(IPM)                                                 006560
      YZ(I+1)=-YZ(I)                                                    006570
10    CONTINUE                                                          006580
11    CONTINUE                                                          006590
C                                                                       006600
C     MOVE ALMOST IDENTICAL ZEROS ADJACENT                              006610
C                                                                       006620
C                                                                       006630
      DO 14 I=1,NNN                                                     006640
      MMM=I+1                                                           006650
      DO 13 JM=MMM,M                                                    006660
      IF ((ABS(XZ(I)-XZ(JM)).GT.E).OR.(ABS(YZ(I)-YZ(JM)).GT.E)) GO TO 13006670
      IIM=JM-I                                                          006680
      DO 12 JJM=1,IIM                                                   006690
      IPM=JM-JJM                                                        006700
      XZ(IPM+1)=XZ(IPM)                                                 006710
12    YZ(IPM+1)=YZ(IPM)                                                 006720
13    CONTINUE                                                          006730
14    CONTINUE                                                          006740
      RETURN                                                            006750
```

```
      END                                                                006760


      SUBROUTINE SMULR (COE,N1,ROOTR,ROOTI)                              006770
C                                                                        006780
C     THIS SUBROUTINE IS USED IN FACTORING THE INPUT NUMERATOR AND       006790
C     DENOMINATOR POLYNOMIALS WHEN THE INPUT TRANSFER FUNCTION IS IN     006800
C     POLYNOMIAL FORM.                                                   006810
C                                                                        006820
      DIMENSION COE(1),ROOTR(1),ROOTI(1)                                 006830
      N4=0                                                               006840
      I=N1+1                                                             006850
1     IF (COE(I)) 3,2,3                                                  006860
2     N4=N4+1                                                            006870
      ROOTR(N4)=0.0                                                      006880
      ROOTI(N4)=0.0                                                      006890
      I=I-1                                                              006900
      IF (N4-N1) 1,20,1                                                  006910
3     CONTINUE                                                           006920
4     AXR=0.8                                                            006930
      AXI=0.0                                                            006940
      L=1                                                                006950
      N3=1                                                               006950
      ALP1R=AXR                                                          006970
      ALP1I=AXI                                                          006980
      M=1                                                                006990
      GO TO 25                                                           007000
5     BET1R=TEMR                                                         007010
```

```
      BET1I=TEMI                                                        007020
      AXR=0.85                                                          007030
      ALP2R=AXR                                                         007040
      ALP2I=AXI                                                         007050
      M=2                                                               007060
      GO TO 26                                                          007070
    6 BET2R=TEMR                                                        007080
      BET2I=TEMI                                                        007090
      AXR=0.9                                                           007100
      ALP3R=AXR                                                         007110
      ALP3I=AXI                                                         007120
      M=3                                                               007130
      GO TO 26                                                          007140
    7 BET3R=TEMR                                                        007150
      BET3I=TEMI                                                        007160
    8 TE1=ALP1R-ALP3R                                                   007170
      TE2=ALP1I-ALP3I                                                   007180
      TE5=ALP3R-ALP2R                                                   007190
      TE6=ALP3I-ALP2I                                                   007200
      TEM=TE5*TE5+TE6*TE6                                               007210
      TE3=(TE1*TE5+TE2*TE6)/TEM                                         007220
      TE4=(TE2*TE5-TE1*TE6)/TEM                                         007230
      TE7=TE3+1.0                                                       007240
      TE9=TE3*TE3-TE4*TE4                                               007250
      TE10=2.0*TE3*TE4                                                  007260
      DE15=TE7*BET3R-TE4*BET3I                                          007270
      DE16=TE7*BET3I+TE4*BET3R                                          007280
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15                               007290
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE15                               007300
      TE7=TE9-1.0                                                       007310
```

```
      TE1=TE9*BET2R-TE10*BET2I                                          007320
      TE2=TE9*BET2I+TE10*BET2R                                          007330
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I                               007340
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R                               007350
      TE15=DE15*TE3-DE16*TE4                                            007360
      TE16=DE15*TE4+DE16*TE3                                            007370
      TE1=TE13*TE13-TE14*TE14-4.0*(TE11*TE15-TE12*TE16)                 007380
      TE2=2.0*TE13*TE14-4.0*(TE12*TE15+TE11*TE16)                       007390
      TEM=SQRT(TE1*TE1+TE2*TE2)                                         007400
      IF (TE1) 9,9,10                                                   007410
    9 TE4=SQRT(0.5*(TEM-TE1))                                           007420
      TE3=0.5*TE2/TE4                                                   007430
      GO TO 13                                                          007440
   10 TE3=SQRT(0.5*(TEM+TE1))                                           007450
      IF (TE2) 11,12,12                                                 007460
   11 TE3=-TE3                                                          007470
   12 TE4=0.5*TE2/TE3                                                   007480
   13 TE7=TE13+TE3                                                      007490
      TE8=TE14+TE4                                                      007500
      TE9=TE13-TE3                                                      007510
      TE10=TE14-TE4                                                     007520
      TE1=2.0*TE15                                                      007530
      TE2=2.0*TE16                                                      007540
      IF (TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10) 14,14,15                   007550
   14 TE7=TE9                                                           007560
      TE8=TE10                                                          007570
   15 TEM=TE7*TE7+TE8*TE8                                               007580
      TE3=(TE1*TE7+TE2*TE8)/TEM                                         007590
      TE4=(TE2*TE7-TE1*TE8)/TEM                                         007600
      AXR=ALP3R+TE3*TE5-TE4*TE6                                         007610
```

```
      AXI=ALP3I+TE3*TE6+TE4*TE5                                          007620
      ALP4R=AXR                                                          007630
      ALP4I=AXI                                                          007640
      M=4                                                                007650
      GO TO 26                                                           007660
   16 IF (ABS(HELL)+ABS(BELL)-1.0E-20) 19,19,17                          007670
   17 TE7=ABS(ALP3R-AXR)+ABS(ALP3I-AXI)                                  007680
      IF (TE7/(ABS(AXR)+ABS(AXI))-1.0E-7) 19,19,18                       007690
   18 N3=N3+1                                                            007700
      ALP1R=ALP2R                                                        007710
      ALP1I=ALP2I                                                        007720
      ALP2R=ALP3R                                                        007730
      ALP2I=ALP3I                                                        007740
      ALP3R=ALP4R                                                        007750
      ALP3I=ALP4I                                                        007760
      BET1R=BET2R                                                        007770
      BET1I=BET2I                                                        007780
      BET2R=BET3R                                                        007790
      BET2I=BET3I                                                        007800
      BET3R=TEMR                                                         007810
      BET3I=TEMI                                                         007820
      IF (N3-100) 8,19,19                                                007830
   19 N4=N4+1                                                            007840
      ROOTR(N4)=ALP4R                                                    007850
      ROOTI(N4)=ALP4I                                                    007860
      N3=0                                                               007870
      IF (N4-N1) 21,20,20                                                007880
   20 RETURN                                                             007890
   21 IF (ABS(ROOTI(N4))-1.0E-5) 4,4,22                                  007900
   22 GO TO (23,4), L                                                    007910
```

```
23      AXR=ALP1R                                                       007920
        AXI=-ALP1I                                                      007930
        ALP1I=-ALP1I                                                    007940
        M=5                                                             007950
        GO TO 26                                                        007960
24      BET1R=TEMR                                                      007970
        BET1I=TEMI                                                      007980
        AXR=ALP2R                                                       007990
        AXI=-ALP2I                                                      008000
        ALP2I=-ALP2I                                                    008010
        M=6                                                             008020
        GO TO 26                                                        008030
25      BET2R=TEMR                                                      008040
        BET2I=TEMI                                                      008050
        AXR=ALP3R                                                       008060
        AXI=-ALP3I                                                      008070
        ALP3I=-ALP3I                                                    008080
        L=2                                                             008090
        M=3                                                             008100
26      TEMR=COE(1)                                                     008110
        TEMI=0.0                                                        008120
        DO 27 I=1,N1                                                    008130
        TE1=TEMR*AXR-TEMI*AXI                                           008140
        TEMI=TEMI*AXR+TEMR*AXI                                          008150
27      TEMR=TE1+COE(I+1)                                               008160
        HELL=TEMR                                                       008170
        BELL=TEMI                                                       008180
        IF (N4) 28,30,28                                                008190
28      DO 29 I=1,N4                                                    008200
        TEM1=AXR-ROOTR(I)                                               008210
```

```
      TEM2=AXI-ROOTI(I)                                                 008220
      TE1=TEM1*TEM1+TEM2*TEM2                                           008230
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1                                     008240
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1                                    008250
29    TEMR=TE2                                                          008260
30    GO TO (5,5,7,16,24,25), N                                         008270
      END                                                               008280


      SUBROUTINE ANG1                                                   008290
C                                                                       008300
C     THIS SUBROUTINE COMPUTES MAGNITIUDE AND PHASE ANGLE               008310
C     (SEE SUBROUTINE GANG1)                                            008320
C     WHERE "MAGNITUDE" IS                                              008330
C     GAIN = GA*(!T-P(1)!*!T-P(2)!*...*!T-P(N)!)                        008340
C              /(!T-Z(1)!*!T-Z(2)!*...*!T-Z(M)!)                        008350
C     NOTE: GA USED HERE IS INVERSE OF INPUT VALUE                      008360
C     !T-P(I)!=SQRT(A**2+B**2)                                          008370
C                                                                       008380
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME    008390
      COMMON/ANGSS/A,B,ANG,PI                                           008400
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                            008410
      GANG=0.                                                           008420
      GAIN=GA                                                           008430
      IF(TIME.NE.0.) GAIN=GAIN*(2.71828183**(TIME*X))                   008440
      FX=0.                                                             008450
      FY=-TIME                                                          008460
      DO 1 I=1,N                                                        008470
```

```
      A=X-XP(I)                                                    008480
      B=Y-YP(I)                                                    008490
      G=A*A+B*B                                                    008500
      IF (G .GE. E*E) GO TO 5                                      008510
      J=5                                                          008520
      GO TO 1                                                      008530
5     FX=FX+B/G                                                    008540
      FY=FY-A/G                                                    008550
      CALL BANG                                                    008560
      GANG=GANG-ANG                                                008570
1     GAIN=GAIN*SQRT(G)                                            008580
      IF (M) 4,4,2                                                 008590
2     DO 3 I=1,M                                                   008600
      A=X-XZ(I)                                                    008610
      B=Y-YZ(I)                                                    008620
      G=A*A+B*B                                                    008630
      IF(G .GE. E*E) GO TO 6                                       008640
      GAIN = 0                                                     008650
      GO TO 3                                                      008660
6     FX=FX-B/G                                                    008670
      FY=FY+A/G                                                    008680
      CALL BANG                                                    008690
      GANG=GANG+ANG                                                008700
      GAIN=GAIN/SQRT(G)                                            008710
3     CONTINUE                                                     008720
4     ANG=GANG-PI*SIG-TIME*Y                                       008730
      CALL BOX                                                     008740
      GANG=ANG                                                     008750
      RETURN                                                       008760
      END                                                          008770
```

```
*      SUBROUTINE DSP           DISPOSE PLOT FILE                          008780
*                                                                          008790
*           AFIT VERSION FOR SCOPE 3.4  APRIL 74                           008800
*                                                                          008810
*           CALL DSP(ID)                                                   008820
*           CALL DSP(ID,LAST)                                              008830
*                                                                          008840
*           WHERE  ID = EITHER: A TWO CHARACTER (HOLLERITH CONSTANT)       008850
*                               GIVING THE SCOPE 3.4 TERMINAL ID           008860
*                         OR: AN INTEGER CONSTANT GIVING THE               008870
*                             OLD TERMINAL ID. SEE THE IDTABLE BELOW       008880
*                             FOR CORRESPONDENCE.                          008890
*                                                                          008900
*                LAST = AN OPTIONAL 2ND ARGUMENT. IF LAST IS NOT USED OR   008910
*                       IF IT HAS ZERO VALUE, THE PLOT FILE IS RE-OPENED   008920
*                       AFTER THE DISPOSE AND A BANNER IS DRAWN WITH       008930
*                       THE JOBNAME, DATE AND TIME.                        008940
*                       IF LAST .NE.0 THE PLOT FILE IS NOT RE-OPENED.      008950
*                                                                          008960
*           IN THIS VERSION OF DSP AN ENDING BANNER IS NOT DRAWN           008970
*           BEFORE THE DISPOSE.                                            008980
*                                                                          008990
          IDENT  DSP                                                      009000
          LIST M,S                                                        009010
          SPACE  1                                                        009020
CALL      MACRO  SUB,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,TRAC                  009030
          LOCAL  SKIP,PARLST                                              009040
          SPACE  1                                                        009050
          IF     -DEF,SUB,1                                               009060
          EXT    SUB                                                      009070
```

```
                EQ      SKIP                                            009080
PARLST          BSS     0                                               009090
                IFC     NE,**A1*                                        009100
                VFD     60/A1                                           009110
                IFC     NE,**A2*                                        009120
                VFD     60/A2                                           009130
                IFC     NE,**A3*                                        009140
                VFD     60/A3                                           009150
                IFC     NE,**A4*                                        009160
                VFD     60/A4                                           009170
                IFC     NE,**A5*                                        009180
                VFD     60/A5                                           009190
                IFC     NE,**A6*                                        009200
                VFD     60/A6                                           009210
                IFC     NE,**A7*                                        009220
                VFD     60/A7                                           009230
                IFC     NE,**A8*                                        009240
                VFD     60/A8                                           009250
                IFC     NE,**A9*                                        009260
                VFD     60/A9                                           009270
                IFC     NE,**A10*                                       009280
                VFD     60/A10                                          009290
                ENDIF                                                   009300
                DATA    0                                               009310
 SKIP           SA1     PARLST                                          009320
+               RJ      SUB                                             009330
                IFC     EQ,**TRAC*,1                                    009340
-               VFD     12/*-TRACE.,18/TRACE.                           009350
                IFC     NE,**TRAC*,1                                    009360
-               VFD     12/*-TRACE.,18/TRAC                             009370
```

```
CALL    ENDM                                                    009380
        SPACE   1                                               009390
        MACRO   TRACEM,TRAC,NAME,INTNAME                        009400
        SPACE   1                                               009410
        ENTRY   NAME                                            009420
X       MICRO   1,,=NAME=           REMOVE POSSIBLE =           009430
SKP1    IFC     NE,*TRAC*TRACE,*                                009440
        IF      -DEF,TRACE.,1                                   009450
TRACE.  BSS     0                                               009460
SKP1    ENDIF                                                   009470
TRAC    VFD     42/7L"X"                                        009480
        VFD     18/NAME                                         009490
        IF      -DEF,TEMPA0.,1                                  009500
TEMPA0. BSS     1                                               009510
        IFC     NE,*NAME*INTNAME*,1                             009520
NAME    BSS     0                                               009530
        IFC     EQ,**INTNAME*,1                                 009540
        DATA    0                                               009550
TRACEM  ENDM                                                    009560
        SPACE   1                                               009570
FTNLNK  MACRO   NOARG               NOARG LIMIT IS 6.           009580
        SPACE   1                                               009590
        IFLE NOARG,6                                            009600
        SX6  A0                     REVISED FOR FORTRAN         009610
        SA0  A1                     EXTENDED LINKEAGES.         009620
        SA6  TEMPA0.                SAVE A0.                    009630
        IFC  NE,**NOARG*                                        009640
        IFNE NOARG,0                                            009650
        IFGE NOARG,1                                            009660
        SB1  X1                     ARG1.                       009670
```

```
          IFGE  NOARG,2                                                  009680
          SA1   A0+1                                                     009690
          SB2   X1                    ARG2.                              009700
          IFGE  NOARG,3                                                  009710
          SA1   A0+2                                                     009720
          SB3   X1                    ARG3.                              009730
          IFGE  NOARG,4                                                  009740
          SA1   A0+3                                                     009750
          SB4   X1                    ARG4.                              009760
          IFGE  NOARG,5                                                  009770
          SA1   A0+4                                                     009780
          SB5   X1                    ARG5.                              009790
          IFGE  NOARG,6                                                  009800
          SA1   A0+5                                                     009810
          SB6   X1                    ARG6.                              009820
          ENDIF                                                          009830
          IFGT  NOARG,6                                                  009840
*********************************************************************    009850
*                                                                   *    009860
 ***THIS MACRO WILL PROCESS A MAX. OF SIX ARGUMENTS*****************     009870
*                                                                   *    009880
*********************************************************************    009890
          ENDIF                                                          009900
FTNLNK    ENDM                                                           009910
          SPACE  1                                                       009920
          TRACEM OSP                                                     009930
          SPACE  1                                                       009940
          FTNLNK  2                                                AFIT-009950
          SPACE  1                                                       009960
          MX0  12                   ..                                   009970
```

```
                SA4   B1                    ..                                  009980
                BX3   X0*X4                 EXTRACT TERMINAL ID.                009990
                NZ      X3,ARGC     JUMP IF NOT INTEGER ID                      010000
                MX0     56          MASK INTEGER ID                             010010
                BX3     -X0*X4                                                  010020
                SA3     X3+IDTABLE  GET NEW CHARACTER ID FROM TABLE             010030
ARGC            SA2   DISP                  ..                          AFIT-010040
                MX0   36                    ..                                  010050
                LX3   24                    POSITION TERMINAL ID.               010060
                BX6   X0*X2                 CLEAR DISPOSE COMPLETION BIT.       010070
                IX6   X6+X3                 INSERT TERMINAL NUMBER.             010080
                SA6   DISP                  ..                                  010090
                MX7     0                                                       010100
                EQ      B0,B2,GO    JUMP IF ONLY ONE ARG                        010110
                SA3     B2                                                      010120
                BX7     X3                                                      010130
GO              SA7     LAST        STORE 2ND ARG IN LAST                       010140
                SPACE   1                                                       010150
                SA4   JOBNAME               ..                                  010160
                NZ    X4,ID5                BYPASS IF WE ALREADY HAVE JOBNAME.  010170
                SPACE   1                                                       010180
                SA5   JOBID                 SET UP PP CALL                      010190
                BX7   X5                    TO GET JOBNAME.                     010200
ID3             SA4   1                     ..                                  010210
                NZ    X4,ID3                LOOP IF RA+1 IS BUSY.               010220
                SA7   1                     ISSUE REQUEST.                      010230
ID3A            SA4   1                     ..                                  010240
                NZ    X4,ID3A               LOOP UNTIL ACCEPTED.                010250
ID4             SA5   JOBNAME               ..                                  010260
                NZ    X5,ID5                CONTINUE WHEN JOBNAME RETURNED.     010270
```

```
          RJ   RELEAS              GO INTO RECALL.                          010280
          EQ   ID4                 ..                                       010290
 ID5      EQ   *+1                 ..                                       010300
          SPACE  1                                                          010310
*  DELETE CARDS DSP00033 THRU DSP00039 TO OMIT ENDING BANNER                010320
 DISP     BSS  0                                                           010330
          SPACE  1                                                          010340
          DISPOSE  PLOT,PT=IXX,RECALL                                       010350
          SPACE  1                                                          010360
          SA3    LAST           OMIT OPEN BANNER                          010370
          NZ     X3,EXIT          IF LAST.NE.0                            010380
          CALL PLOT,=0.25,=10.0,=3                                          010390
          CALL PLOT,=0.25,=8.555,=2                                         010400
          CALL PLOT,=0,=8.555,=2                                            010410
          CALL SYMBOL,=0.1875,=8.805,=0.105,OPNPLT,=90.0,=9                 010420
          CALL PLOT,=0,=0,=3                                                010430
          CALL GILSID,JOBNAME                                               010440
          SPACE  1                                                          010450
 EXIT     SA5  TEMPA0.             ..                                AFIT-010460
          SA0  X5                  RESET A0                                 010470
          EQ   DSP                 AND EXIT.                                010480
          SPACE  1                                                          010490
 RELEAS   DATA 0                   ..                                       010500
          SX6  2203148             .SET RCL (RELEASE CP)                    010510
          LX6  42                  .LEFT JUSTIFY                            010520
          SA6  1                   .SET IN RA+1 (RECALL)                    010530
          EQ   RELEAS              ..                                       010540
          SPACE  1                                                          010550
*ENDDSP   DIS  1, END  DSP    **DELETE THIS CARD FOR AFIT-DSP          AFIT-010560
 OPNPLT   DIS  1,OPEN PLOT                                                  010570
```

```
JOBID    VFD  18/3HPID,24/0,18/JOBNAME                                    010580
JOBNAME  DATA 0                                                           010590
LAST     DATA   0                                                         010600
IDTABLE  DATA   2LAB        T0  ASD/AD  B675 (NO ONLINE PLOTTER)          010610
         DATA   2LBA        T1  AFFDL   B45                              010620
         DATA   2LAD        T2  ASD/ENF B17                              010630
         DATA   2LAC        T3  ASD/XR  B52                              010640
         DATA   2LAE        T4  AFAPL   B18                              010650
         DATA   2LAF        T5  AMRL    B441                             010660
         DATA   2LBC        T6  AFWAL   B450                             010670
         DATA   2LBB        T7  AFIT    B640                             010680
         DATA   2LBD        T8  AFAL    B22                              010690
         END                                                              010700
      OVERLAY(CCPCSD,1,1)                                                 010710
      PROGRAM RJOT11                                                      010720
C                                                                         010730
C     THIS OVERLAY SETS UP THE CALCOMP PLOT                               010740
C                                                                         010750
      COMMON /MAIN01/BOUND,DZW(8),FPN,IFOLD,IFPN,ISTAM,ITIT1,ITHES,RAD    010760
     +  ,THESI(8),ZETA,MULT                                              010770
      COMMON /MAIN012/GANE,NOPLOT,GTOL,IIV,LL,V,XAX,YAX,DEL(50),          010780
     +  XDISL(50),XDISR(50),YDISD(50),YDISU(50),XOT,YOT,CB                010790
      COMMON/ANGR2/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME      010800
      COMMON/ANGSS/A,B,ANG,PI                                             010810
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                              010820
      COMMON /ANGJU/ AA,BB,CC,DD,D,DA, JK,SANG,CA                         010830
      LOGICAL NOPLOT                                                      010840
      DIMENSIONRZ2(50),IID(50),JJD(50),XOTVAL(5)                          010850
      DATA XOTVAL /6.,13.,19.,23.,30./                                    010860
C                                                                         010870
```

```
C     SKIP ALL PLOTTING AND PLOT SETUP IF THIS IS TO BE                  010880
C     A LOCUS ON AN EXISTING PLOT                                        010890
C                                                                        010900
      IF(MULT .NE. 0) GO TO 69                                           010910
C                                                                        010920
C    SCALE S-PLANE BOUNDARIES                                            010930
C                                                                        010940
      AA=AA*BOUND                                                        010950
      BB=BB*BOUND                                                        010960
      CC=CC*BOUND                                                        010970
      DD=DD*BOUND                                                        010980
      AAB=AA-CC                                                          010990
      BBD=BB-DD                                                          011000
C                                                                        011010
C    DETERMINE PLOT SIZE AND ORIENTATION AND PRINT OUTSIDE TITLE         011020
C                                                                        011030
      YOT=9.                                                             011040
      XOT = XOTVAL(MAX0(MIN0(4,IFOLD),0)+1)                              011050
      IF (IFOLD.NE.0) GO TO 20                                           011060
      IF (BBD.GT.AAB) GO TO 20                                           011070
      XOT=9.                                                             011080
      YOT=6.                                                             011090
      IF (NOPLOT) GO TO 21                                               011100
      IF(ITHES .LT. 0) CALL SYMBOL(9.375,6.0,.105,THESI,-90.,80)         011110
      GO TO 21                                                           011120
 20   IF (NOPLOT) GO TO 21                                               011130
      IF(ITHES .LT. 0) CALL SYMBOL(0,9.375,.105,THESI,0,80)              011140
C                                                                        011150
C     READJUST BB, CC, AND DD TO FIT THE LENGTH AND WIDTH OF THE PLOT    011160
C                                                                        011170
```

```
21      WV=(YOT-2.)/(XOT-1.)                                          011180
        CBZ=WV*AAB                                                    011190
        CC=AA-BBD/WV                                                  011200
        IF (BBD.GE.CBZ) GO TO 22                                      011210
        BB=BB+.5*(CBZ-BBD)                                            011220
        DD=DD-.5*(CBZ-BBD)                                            011230
        CC=AA-AAB                                                     011240
C                                                                     011250
C       COMPUTE SCALE FACTOR AND RECIPROCA.                           011260
C                                                                     011270
C     SCALE FACTOR IN INCHES PER UNIT                                 011280
22      V=(XOT-1.)/(AA-CC)                                            011290
C     RECIPROCAL SCALE FACTOR IN UNITS PER INCH                       011300
        VW=(AA-CC)/(XOT-1.)                                           011310
        XAA=XOT-.5                                                    011320
        YBB=YOT-.75                                                   011330
C                                                                     011340
C       NUMBER OF DASHES IN X AND Y AXES                              011350
C                                                                     011360
        IAY=(YOT-2.)*10.+E                                            011370
        IAX=(XOT-1.)*10.+E                                            011380
C                                                                     011390
C       COMPUTE COORDINATES OF (0,0)                                  011400
C                                                                     011410
        YAX=.5-V*CC                                                   011420
        XAX=1.25-V*DD                                                 011430
        IF ((LL.EQ.4).OR.(LL.EQ.5)) GO TO 750                         011440
        IF(NOPLOT) GO TO 25                                           011450
C                                                                     011460
C       DRAW OUTLINE                                                  011470
```

```
C                                                                       011480
      CALL PLOT (0.0,0.0,3)                                             011490
      CALL PLOT (0.0,YOT,2)                                             011500
      CALL PLOT (XOT,YOT,2)                                             011510
      CALL PLOT (XOT,0.0,2)                                             011520
      CALL PLOT (0.0,0.0,2)                                             011530
      XYAX = AINT(10.*(YAX-.5))                                         011540
      YXAX = AINT(10.*(XAX-1.25))                                       011550
C                                                                       011560
C     CENTER AND PRINT FIGURE NUMBER                                    011570
C                                                                       011580
      CBZ=XOT                                                           011590
      IF(IFPN.EQ.0) GO TO 223                                           011600
      AAB=CBZ/2.-1.54                                                   011610
      CALL SYMBOL (AAB,.20,.14,22HFIGURE    - ROOT LOCUS,0.0,22)        011620
      CALL NUMBER (AAB+.98,.20,.14,FPN,0.0,-1)                          011630
C                                                                       011640
C     CENTER AND PRINT INSIDE TITLE                                     011650
C                                                                       011660
 223  IF(ITITL.LE.0)GO TO 24                                            011670
      TITSIZ = .175                                                     011680
      IF(ITITL*.175 .GT. XOT-.25) TITSIZ=(XOT-.25)/ITITL                011690
      AAB = CBZ/2.-(TITSIZ*.5*ITITL)                                    011700
      CALL SYMBOL(AAB,YOT-.5,TITSIZ,DZW,0,ITITL)                        011710
C                                                                       011720
C     PRINT SCALE FACTOR AND BOX AROUND IT                              011730
C                                                                       011740
24    CBZ=VW                                                            011750
      JK=4                                                              011760
      CALL DNTER (CBZ,JK)                                               011770
```

```
      ASP=CBZ*.070                                                    011780
      CBZ=VW                                                          011790
      CALL CNTER (CBZ)                                                011800
      ASP=ASP+CBZ*.070                                                011810
      CEA=CBZ                                                         011820
      ASP=XOT-1.73-ASP                                                011830
      CALL PLOT (ASP,1.03,3)                                          011840
      CALL PLOT (XOT-.5,1.03,2)                                       011850
      CALL PLOT (XOT-.5,1.165,2)                                      011860
      CALL PLOT (ASP,1.165,2)                                         011870
      CALL PLOT (ASP,1.03,2)                                          011880
      ASP=ASP+.03                                                     011890
      CALL SYMBOL (ASP,1.055,.07,7HSCALE- ,0.0,7)                     011900
      ASP=ASP+.49                                                     011910
      CALL NUMBER (ASP,1.055,.07,VW,0.0,JK)                           011920
      ASP=XOT-1.27                                                    011930
      CALL SYMBOL (ASP,1.055,.07,11H UNITS/INCH,0.0,11)               011940
C                                                                     011950
C     SEARCH ALONG RADIAL LINE FOR SOLUTION AT DESIRED ZETA           011960
C     AND COMPUTE GAIN FACTOR.                                        011970
C                                                                     011980
 25   IF (.NOT.((LL.EQ.3).OR.(LL.EQ.7)).OR.(ZETA.GT..91)) GO TO 31   011990
      XWORK = -ZETA                                                   012000
      YWORK=SQRT(1.-ZETA*ZETA)                                        012010
      XX = X = XWORK*RAD                                              012020
      YY = Y = YWORK*RAD                                              012030
      CALL GANG1                                                      012040
      D = DI = DEL(1)                                                 012050
      IF(D .EQ. 0) D = DI = AMAX1(.1,.04*VW)                          012060
 26   X=XX+D*XWORK                                                    012070
```

```
      Y=YY+D*YWORK                                                    012080
      CB=GANG                                                         012090
      CALL GANG1                                                      012100
      IF (CB*GANG) 28,29,27                                           012110
   27 IF ((X .LT.00).OR.( Y.GT.BB)) GO TO 31                          012120
      XX=X                                                            012130
      YY=Y                                                            012140
      GO TO 26                                                        012150
   28 IF(ABS(GANG) .GT. 2.)GO TO 25                                   012160
      GANG = CB                                                       012170
      D=D/4.                                                          012180
      IF (( DI/256.-D).LT.0.) GO TO 26                                012190
   29 CALL ANG1                                                       012200
      J=1                                                             012210
      GANE=GAIN                                                       012220
      GTOL=.1*GANE                                                    012230
   31 IF (NOPLOT) GO TO 32                                            012240
      IF (GTOL.EQ.0.) GO TO 32                                        012250
C                                                                     012260
C     COMPUTE SIZE OF GAIN BOX                                        012270
C                                                                     012280
       CBZ=GANE                                                       012290
      CALL CNTER (CBZ)                                                012300
      JK=2                                                            012310
      XWORK = CBZ                                                     012320
      CBZ=GANE                                                        012330
      CALL CNTER (CBZ,JK)                                             012340
      CBZ = CBZ+XWORK                                                 012350
      XWORK=.825                                                      012360
      IF(GA .NE. 1.) XWORK=.85                                        012370
```

```
      ASP=XWORK+CBZ*.070                                                012380
C                                                                       012390
C     PLOT GAIN BOX AND GAIN OF INTEREST                                012400
C                                                                       012410
      CALL PLOT (.5,1.03,3)                                             012420
      CALL PLOT (ASP,1.03,2)                                            012430
      CALL PLOT (ASP,1.165,2)                                           012440
      CALL PLOT (.5,1.165,2)                                            012450
      CALL PLOT (.5,1.03,2)                                             012460
      CALL SYMBOL (.565,1.09,.07,2,0.0,-1)                              012470
      CALL SYMBOL (.6,1.055,.07,2H K,0.0,2)                             012480
      IF(GA.NE.1.)CALL SYMBOL (.74,1.055,.035,1HN,0.0,1)                012490
      CALL SYMBOL(XWORK-.08,1.055,.07,1H=,0,1)                          012500
      CALL NUMBER(XWORK,1.055,.07,GANE,0,JK)                            012510
32    IF (ISTAM.LT.0) GO TO 50                                          012520
      IF (NOPLOT) GO TO 320                                             012530
C                                                                       012540
C     DRAW BOX FOR TRANSFER FUNCTION                                    012550
C                                                                       012560
      CALL PLOT (.5,.5,3)                                               012570
      CALL PLOT (XOT-.5,.5,2)                                           012580
      CALL PLOT (XOT-.5,1.,2)                                           012590
      CALL PLOT (.5,1.,2)                                               012600
      CALL PLOT (.5,.5,2)                                               012610
320   IF (ISTAM.GT.0) GO TO 48                                          012620
      IF (NOPLOT) GO TO 50                                              012630
C                                                                       012640
C     SET UP TO PRINT TRANSFER FUNCTION                                 012650
C    DETERMINE SIZE OF TRANSFER FUNCTION FOR PLOT                       012660
C                                                                       012670
```

```
      JROOT=0                                                          012680
      IIM=0                                                            012690
      KKV=0                                                            012700
      KROOT=1                                                          012710
      JJN=N                                                            012720
      DO 33 I=1,N                                                      012730
      YDISD(I)=YP(I)                                                   012740
33    XDISL(I)=XP(I)                                                   012750
34    CALL GROUP (XDISL,XDISR,YDISD,YDISJ,IID,KKV,JJN)                 012760
C                                                                      012770
C     IID= 0 - DAMPING TERM IS ZERO                                    012780
C     IID= 1 - ROOT AT ORIGIN                                          012790
C     IID= 2 - REPEATED ROOT AT ORIGIN                                 012800
C     IID= 4 - REAL ROOT                                               012810
C     IID= 5 - REPEATED REAL ROOT                                      012820
C     IID= 6 - PURE IMAGINARY COEFFICIENT                              012830
C     IID= 7 - DAMPING TERM COEFFICIENT                                012840
C     IID= 8 - ZEROETH ORDER COEFFICIENT OF COMPLEX ROOT               012850
C                                                                      012860
      XWORK=0                                                          012870
      DO 35 I=KROOT,KKV                                                012880
      AAB=IID(I)                                                       012890
C                                                                      012900
C     AT ORIGIN                                                        012910
      IF ( AAB  .LT.3) GO TO 35                                        012920
C                                                                      012930
C     COMPLEX PAIR                                                     012940
      IF ( AAB .EQ. 8.) AAB=0                                          012950
C                                                                      012960
C     IMAGINARY PAIR                                                   012970
```

```
      IF ( AAB .EQ. 6.) AAB=5.                                          012980
      CBZ=XDISR(I)                                                      012990
      CALL CNTER (CBZ)                                                  013000
      ASP=CBZ                                                           013010
      CBZ=XDISR(I)                                                      013020
      JK=3                                                              013030
      CALL CNTER (CBZ,JK)                                               013040
      JJD(I)=JK                                                         013050
      AAB=AAB+CBZ+ASP                                                   013060
C                                                                       013070
C     # OF SPACES FOR COEFFICIENT "XDISR(I)"                            013080
      RZD(I)=CBZ+ASP                                                    013090
 35   XWORK=XWORK+AAB                                                   013100
      EEA=XWORK                                                         013110
      IF (IIM.EQ.1) GO TO 37                                            013120
      IIM=1                                                             013130
C     LENGTH OF DENOMINATOR                                             013140
      EEP=XWORK                                                         013150
      EEA=0.                                                            013160
      JROOT=KKV                                                         013170
      IF (M.EQ.0) GO TO 37                                              013180
      JJN=M                                                             013190
      KROOT=KKV+1                                                       013200
      DO 36 I=1,M                                                       013210
      XDISL(I)=XZ(I)                                                    013220
36    YDISD(I)=YZ(I)                                                    013230
      GO TO 34                                                          013240
C     LENGTH OF NUMERATOR                                               013250
37    EEA=EEA+1.                                                        013260
C                                                                       013270
```

```
C     DETERMINE SIZE FOR PLOTTING TIME DELAY IF APPROPRIATE                013280
C                                                                          013290
      IF (TIME.LT.E) GO TO 38                                              013300
      CBZ=-TIME                                                            013310
      CALL CNTER (CBZ)                                                     013320
      TIMA=CBZ                                                             013330
      CBZ=-TIME                                                            013340
      JK=3                                                                 013350
      CALL CNTER (CBZ,JK)                                                  013360
      TIMA=TIMA+CBZ                                                        013370
      NID=JK                                                               013380
      EEA=EEA+TIMA+2.                                                      013390
C                                                                          013400
C     SELECT LETTER SIZE                                                   013410
C                                                                          013420
38    ASP=EEP                                                              013430
      IF (EEA.GT.EEP) ASP=EEA                                              013440
      CBZ=ASP*.105                                                         013450
      IF((XOT-1.945).GT.CBZ)GO TO 40                                       013460
      CBZ=ASP*.07                                                          013470
      IF((XOT-1.945).GT.CBZ)GO TO 40                                       013480
      IF((XOT-1.).GT.CBZ)GO TO 39                                          013490
      CBZ=.035*ASP                                                         013500
C     DETERMINE IF TRANSFER FUNCTION FITS BOX                              013510
      IF((XOT-1.).LT.CBZ)GO TO 50                                          013520
C                                                                          013530
C     USE FULL LABEL                                                       013540
C     THERE IS ROOM OVER THE TRANSFER FUNCTION                             013550
C                                                                          013560
 39   VIX=(XOT-3.57)/2.                                                    013570
```

```
      CALL SYMBOL (VIX,.843,.105,34HTHE OPEN LOOP TRANSFER FUNCTION IS,0013580
     1.0,34)                                                          013590
      VIY=.67                                                         013600
      VIX=XOT/2.                                                      013610
      AAB=VIX-CBZ/2.                                                  013620
      IF (AAB.LT..5) GO TO 50                                         013630
      GO TO 41                                                        013640
C                                                                     013650
C     USE CONDENSED LABEL                                             013660
C     PUT IT IN FRONT OF THE TRANSFER FUNCTION                        013670
C                                                                     013680
 40   VIX=(XOT-CBZ)/2.-.472                                           013690
      VIY=.75                                                         013700
      CALL SYMBOL (VIX,.7,.105,8HG(S)H(S)=,0.0,9)                     013710
      VIX=VIX+CBZ/2.+.945                                             013720
C                                                                     013730
C     NOW PRINT TRANSFER FUNCTION                                     013740
C                                                                     013750
41    BBD=CBZ/ASP                                                     013760
      AAB=VIX-CBZ/2.                                                  013770
      CALL PLOT (AAB,VIY,3)                                           013780
      AAB=VIX+CBZ/2.                                                  013790
      CALL PLOT (AAB,VIY,2)                                           013800
      CBZ=BBD                                                         013810
      AAB=VIX-EEA*BBD/2.                                              013820
      VIX=VIX-EEP*BBD/2.                                              013830
      BBD=VIY                                                         013840
      VIY=BBD-1.6*CBZ                                                 013850
      DO 47 I=1,K<V                                                   013860
      JJN=IID(I)                                                      013870
```

```
      IF (JJN.EQ.1.OR.JJN.EQ.2) GO TO 44                                013880
      IF (JJN.EQ.6.OR.JJN.EQ.7) GO TO 42                                013890
      CALL SYMBOL (VIX,VIY,CBZ,2H(S,0.0,2)                              013900
      VIX=VIX+2.*CBZ                                                    013910
      IF (JJN.EQ.8.OR.JJN.EQ.0) GO TO 45                                013920
42    IF (XDISR(I).LT.0.) GO TO 43                                      013930
      CALL SYMBOL (VIX,VIY,CBZ,1H+,0.0,1)                               013940
      VIX=VIX+CBZ                                                       013950
43    JK=JJD(I)                                                         013960
      EEP=XDISR(I)                                                      013970
      CALL NUMBER (VIX,VIY,CBZ,EEP,0.0,JK)                              013980
      VIX=VIX+CBZ*RZD(I)                                                013990
      IF (JJN.EQ.8) GO TO 44                                            014000
      CALL SYMBOL (VIX,VIY,CBZ,1H),0.0,1)                               014010
      VIX=VIX+CBZ                                                       014020
      IF (JJN.EQ.5) GO TO 45                                            014030
      GO TO 46                                                          014040
44    CALL SYMBOL (VIX,VIY,CBZ,1HS,0.0,1)                               014050
      VIX=VIX+CBZ                                                       014060
      IF (JJN.EQ.8.OR.JJN.EQ.1) GO TO 46                                014070
45    EEP=2.                                                            014080
      IF (JJN.EQ.5.OR.JJN.EQ.2) EEP=YDISJ(I)+E                          014090
      ASP=.7*CBZ                                                        014100
      EEA=VIY+.6*CBZ                                                    014110
      CALL NUMBER (VIX,EEA,ASP,EEP,0.0,-1)                              014120
      VIX=VIX+.75*CBZ                                                   014130
      IF (JJN.EQ.8) GO TO 42                                            014140
46    IF (I.NE.JROOT) GO TO 47                                          014150
      VIY=.30*CBZ+BBD                                                   014160
      VIX=AAB                                                           014170
```

```
      CALL SYMBOL (VIX,VIY,CBZ,1HK,0.0,1)                                014180
      IF(SIG.EQ.0.) CALL SYMBOL (VIX-CBZ,VIY,CBZ,1H-,0.0,1)              014190
      VIX=VIX+CBZ                                                        014200
47    CONTINUE                                                           014210
C                                                                        014220
C     COMPLETE TRANSFER FUNCTION IF THERE IS A TIME DELAY                014230
C                                                                        014240
      IF (TIME.LT.E) GO TO 50                                            014250
      EEP=.8*CBZ                                                         014260
      CALL SYMBOL (VIX,VIY,EEP,35,0.0,-1)                                014270
      EEA=VIY+.4*CBZ                                                     014280
      BBD=VIX+.2*CBZ                                                     014290
      EEP=.4*CBZ                                                         014300
      CALL SYMBOL (BBD,EEA,EEP,31,0.0,-1)                                014310
      VIX=VIX+CBZ                                                        014320
      VIY=VIY+.6*CBZ                                                     014330
      EEP=CBZ*.7                                                         014340
      EEA=-TIME                                                          014350
      CALL NUMBER (VIX,VIY,EEP,EEA,0.0,NID)                              014360
      VIX=VIX+.70*TIMA*CBZ                                               014370
      CALL SYMBOL (VIX,VIY,EEP,46,0.0,-1)                                014380
      GO TO 50                                                           014390
C                                                                        014400
C     PRINT 3 LINES IN PLACE OF TRANSFER FUNCTION                        014410
C                                                                        014420
48    VIY=.849                                                           014430
      TITSIZ  = AMIN1(.105,(XOT-1.)/ISTAM)                               014440
      VIX=(XOT-ISTAM*TITSIZ)*.5                                          014450
      WRITE(7,200)                                                       014460
      DO 49 I=1,3                                                        014470
```

```
      READ(5,181)(RZQ(IIM),IIM=1,8)                                          014480
      WRITE(6,179)(RZQ(IIM),IIM=1,8)                                         014490
      IF (NOPLOT) GO TO 480                                                  014500
      CALL SYMBOL(VIX,VIY,TITSIZ,RZQ,0,ISTAM)                                014510
480   VIY=VIY-.148                                                           014520
48    CONTINUE                                                               014530
C                                                                            014540
C     CHECK FOR REPEATED OPEN-LOOP ZEROES                                    014550
C                                                                            014560
50    IIV=0                                                                  014570
      IF (M.LT.2) GO TO 55                                                   014580
      LQ=1                                                                   014590
      DO 54 I=2,M                                                            014600
      IF (XZ(I).EQ.XZ(I-1).AND.YZ(I).EQ.YZ(I-1)) GO TO 53                    014610
51    IF (LQ.EQ.1) GO TO 54                                                  014620
      ASP=XZ(I-1)*V+YAX                                                      014630
      EEP=YZ(I-1)*V+XAX                                                      014640
      IF ((ASP.GT.(XOT-.5)).OR.(ASP.LT..5).OR.(EEP.LT.1.25).OR.(EEP.GT.(014650
     1YOT-.75))) GO TO 52                                                    014660
      IF (NOPLOT) GO TO 510                                                  014670
C                                                                            014680
C     PLOT ANY REPEATED OPEN-LOOP ZEROES                                     014690
C                                                                            014700
      CALL SYMBOL (ASP,EEP,.035,1,0.0,-1)                                    014710
      CALL SYMBOL (ASP+.055,EEP+.03,.07,3H( ),0.0,3)                         014720
510   BBD=LQ                                                                 014730
      IF (NOPLOT) GO TO 511                                                  014740
      CALL NUMBER (ASP+.125,EEP+.03,.07,BBD,0.0,-1)                          014750
511   IIV=IIV+1                                                              014760
      YDISD(IIV)=EEP+.02                                                     014770
```

```
      YDISU(IIV)=EEP+.11                                                014780
      XDISL(IIV)=ASP+.05                                                014790
      XDISR(IIV)=ASP+.23                                                014800
52    LQ=0                                                              014810
53    LQ=LQ+1                                                           014820
      IF (I.EQ.1) GO TO 51                                              014830
54    CONTINUE                                                          014840
55    IF (NOPLOT) GO TO 75                                              014850
C                                                                       014860
C     PLOT AND LABEL REAL AXIS                                          014870
C     VW=UNITS/IN.                V=1/VW                                014880
C     YYB=YOT-.75                                                       014890
C     XAA=XOT-.5                                                        014900
C     XAX=1.25-V*DD                                                     014910
C     YAX=.5-V*CC                                                       014920
C     CEA=# CIPHERS LEFT OF DECIMAL IN VW                               014930
C                                                                       014940
      CBZ=VW                                                            014950
      JK=2                                                              014960
      CALL DNTER (CBZ,JK)                                               014970
      NID=JK                                                            014980
      ASP=.07*CBZ                                                       014990
      DXAX=1.2                                                          015000
      EXAX=1.34                                                         015010
      DYAX=XAA-CEA*.07-ASP-.07                                          015020
      EYAX=DYAX-.07                                                     015030
      IF (XAX.LT.(1.25).OR.XAX.GT.YBB) GO TO 58                         015040
      IF (YAX.GT.(XAA-.4)) GO TO 55                                     015050
C     DRAW SIGMA                                                        015060
      CALL SYMBOL (XOT-.55,XAX+.05,.14,108,0,-1)                        015070
```

```
      IIV=IIV+1                                                          015080
      XDISR(IIV)=XOT-.4                                                  015090
      XDISL(IIV)=XOT-.57                                                 015100
      YDISU(IIV)=XAX+.13                                                 015110
      YDISD(IIV)=XAX+.04                                                 015120
56    DXAX=XAX-.14                                                       015130
      EXAX=XAX                                                           015140
      XWORK=YAX-XYAX/10.-.01                                             015150
      DO 57 I  =1,IAX                                                    015160
C     DRAW X AXIS                                                        015170
      CALL PLOT (XWORK,XAX,3)                                            015180
      CALL PLOT (XWORK+.02,XAX,2)                                        015190
 57   XWORK = XWORK + .1                                                 015200
C                                                                        015210
C     PLOT AND LABEL IMAGINARY AXIS                                      015220
C                                                                        015230
58    IF (YAX.LT.(.5).OR.YAX.GT.XAA) GO TO 60                            015240
      IIV=IIV+1                                                          015250
C     DRAW J-OMEGA                                                       015260
      CALL SYMBOL (YAX-.13,YOT-.81,.09,37,0.0,-1)                        015270
      CALL SYMBOL (YAX-.113,YOT-.73,.14,114,0,-1)                        015280
      CALL SYMBOL (YAX-.15,YOT-.7,.07,74,0.0,-1)                         015290
      XDISR(IIV)=YAX-.045                                                015300
      XDISL(IIV)=YAX-.2                                                  015310
      YDISU(IIV)=YOT-.7                                                  015320
      YDISD(IIV)=YOT-.82                                                 015330
      DYAX=YAX+.07                                                       015340
      EYAX=YAX                                                           015350
      YWORK = XAX - YYAX/10. - .01 +(IAY-1)*.1                           015360
      DO 59 I  =1,IAY                                                    015370
```

```
C       DRAW Y AXIS                                                     015380
        CALL PLOT(YAX,YWORK+.02,3)                                      015390
        CALL PLOT(YAX,YWORK,2)                                          015400
 59     YWORK = YWORK - .1                                              015410
C                                                                       015420
C       SET UP TO DRAW SCALES                                           015430
C                                                                       015440
60      IF(IFPN.GT.0) FPN=FPN+1.                                        015450
        IVW=BB/VW                                                       015460
        WV=IVW                                                          015470
C      TOTAL # OF UNITS ON + IMAGINARY AXIS                             015480
        CBZ=WV*VW                                                       015490
        CALL CNTER (CBZ)                                                015500
        EEA=CBZ                                                         015510
        IVW=DD/VW                                                       015520
        WV=IVW                                                          015530
C      TOTAL # OF UNITS ON - IMAGINARY AXIS                             015540
        CBZ=WV*VW                                                       015550
        CALL CNTER (CBZ)                                                015560
        IF (CBZ.GT.EEA) EEA=EEA+1.                                      015570
        CEA=EEA*.07+ASP+DYAX-.01                                        015580
C                                                                       015590
C       DRAW Y AXIS SCALE                                               015600
C                                                                       015610
61      WV=IVW                                                          015620
        WVW=WV*VW                                                       015630
        IF (WVW.GT.BB) GO TO 63                                         015640
        WV=XAX+WV                                                       015650
        CALL SYMBOL (EYAX,WV,.035,3,0.0,-1)                             015660
        IF (WVW.EQ.0.) GO TO 62                                         015670
```

```
      WV=WV-.035                                                        015680
      CBZ=WVW                                                           015690
      CALL CNTER (CBZ)                                                  015700
      DTAX=DYAX+(EEA-CBZ)*.07                                           015710
      CALL NUMBER (DTAX,WV,.07,WVW,0.0,NID)                             015720
      IIV=IIV+1                                                         015730
      XDISR(IIV)=CEA                                                    015740
      XDISL(IIV)=DTAX-.01                                               015750
      YDISD(IIV)=WV-.01                                                 015760
      YDISU(IIV)=WV+.08                                                 015770
62    IVW=IVW+1                                                         015780
      GO TO 61                                                          015790
63    IVW=CC/VW                                                         015800
C                                                                       015810
C     DRAW X AXIS SCALE                                                 015820
C                                                                       015830
64    WV=IVW                                                            015840
      WVW=WV*VW                                                         015850
      IF (WVW.GT.AA) GO TO 66                                           015860
      WV=YAX+WV                                                         015870
      CALL SYMBOL (WV,EXAX,.035,3,0.0,-1)                               015880
      IF (WVW.EQ.0.) GO TO 66                                           015890
      CBZ=WVW                                                           015900
      CALL CNTER (CBZ)                                                  015910
      CEA=CBZ*.035+ASP/2.                                               015920
      WV=WV-CEA                                                         015930
      CALL NUMBER (WV,DXAX,.070,WVW,0.0,NID)                            015940
      IIV=IIV+1                                                         015950
      XDISL(IIV)=WV-.01                                                 015960
      XDISR(IIV)=WV+2.*CEA-.01                                          015970
```

```
      YDISD(IIV)=DXAX-.01                                                    015980
      YDISU(IIV)=DXAX+.08                                                    015990
65    IVW=IVW+1                                                              016000
      GO TO 64                                                               016010
C                                                                            016020
C     DRAW ZETA LINE                                                         016030
C                                                                            016040
66    IF ((ZETA.LT.(.09)).OR.(ZETA.GT.(.91))) GO TO 69                       016050
      CALL PLOT (YAX,XAX,3)                                                  016060
      EEA=SQRT(1.-ZETA*ZETA)                                                 016070
      VIY=XAX                                                                016080
      VIX=YAX                                                                016090
      NID=2                                                                  016100
67    VIX=VIX-.25*ZETA                                                       016110
      VIY=VIY+.25*EEA                                                        016120
      IF ((((VIY+.90).GT.YDT).OR.(VIX.LT..8)).AND.(NID.EQ.3)) GO TO 68       016130
      CALL PLOT (VIX,VIY,NID)                                                016140
      NID = (NID+4)/NID                                                      016150
      GO TO 67                                                               016160
C                                                                            016170
C     PRINT ZETA                                                             016180
C                                                                            016190
68    VIY=VIY-.2*EEA                                                         016200
      VIX=VIX-.1*ZETA                                                        016210
      CALL SYMBOL(VIX-.05,VIY-.03,.20,96,0,-1)                               016220
      CALL SYMBOL (VIX+.05,VIY,.095,14=,0.0,1)                               016230
      CALL NUMBER (VIX+.144,VIY,.105,ZETA,0.0,2)                             016240
      IIV=IIV+1                                                              016250
      XDISL(IIV)=VIX-.050                                                    016260
      YDISD(IIV)=VIY-.035                                                    016270
```

```
      YDISU(IIV)=VIY+.11                                                    016280
      XDISR(IIV)=VIX+.52                                                    016290
C                                                                           016300
C     PLOT ANY ZERO THAT COINCIDES WITH A POLE                              016310
C     AND REMOVE BOTH FROM FURTHER CONSIDERATION                            016320
C                                                                           016330
69    IF (M.EQ.0) GO TO 75                                                  016340
      DO 74 I=1,M                                                           016350
      DO 73 IIM=1,N                                                         016360
      IF (ABS(XZ(I)-XP(IIM)).GT.E.OR.ABS(YZ(I)-YP(IIM)).GT.E) GO TO 73      016370
      VIX=XZ(I)*V+YAX                                                       016380
      VIY=YZ(I)*V+XAX                                                       016390
      IF ((VIX.LT..5).OR.(VIX.GT.(XOT-.5)).OR.(VIY.LT.1.25).OR.             016400
     1(VIY.GT.(YOT-.75))) GO TO 695                                         016410
      CALL SYMBOL (VIX,VIY,.07,1,0.0,-1)                                    016420
      CALL SYMBOL (VIX,VIY,.07,4,0.0,-1)                                    016430
695   N=N-1                                                                 016440
      IF (IIM.GT.N) GO TO 71                                                016450
      DO 70 JUNK=IIM,N                                                      016460
      XP(JUNK)=XP(JUNK+1)                                                   016470
70    YP(JUNK)=YP(JUNK+1)                                                   016480
71    M=M-1                                                                 016490
      IF ( I.GT.M) GO TO 75                                                 016500
      DO 72 JUNK=I,M                                                        016510
      XZ(JUNK)=XZ(JUNK+1)                                                   016520
72    YZ(JUNK)=YZ(JUNK+1)                                                   016530
      GO TO 69                                                              016540
73    CONTINUE                                                              016550
74    CONTINUE                                                              016560
75    JK=0                                                                  016570
```

```
  750   CONTINUE                                                        016580
C                                                                       016590
C     FORMATS                                                           016600
C                                                                       016610
  170   FORMAT("0"8A10)                                                 016620
  181   FORMAT(8A10)                                                    016630
  200   FORMAT(".ENTER THREE LINE TITLE")                               016640
        END                                                             016650


        SUBROUTINE ONTER (CBZ)                                          016660
C                                                                       016670
C       COMPUTES THE NUMBER OF CHARACTERS IN INTEGER PART OF CBZ        016680
C       RETURNS ANSWER IN CBZ                                           016690
C                                                                       016700
        RILA=CBZ                                                        016710
        CBZ=1.                                                          016720
        IF (RILA.LT.0.) CBZ=2.                                          016730
        RILA=ABS(RILA)                                                  016740
        CBZ = CBZ + MAX0(IFIX(ALOG10(RILA)+.00000001),0)                016750
        RETURN                                                          016760
        END                                                             016770


        SUBROUTINE ONTER (CBZ,JK)                                       016780
C                                                                       016790
```

```
C       INPUT                                                                   016800
C            CBZ  VALUE                                                         016810
C            JK   MAXIMUM NUMBER OF DECIMAL PLACES                              016820
C       OUTPUT:                                                                 016830
C            CBZ  NUMBER OF DECIMAL PLACES TO PLOT + 1 FOR DECIMAL POINT        016840
C             JK  NECESSARY NUMBER OF DECIMAL PLACES OR -1 FOR NONE             016850
C                                                                               016860
        Q=.000001                                                               016870
        NJK=-1                                                                  016880
        RILA=ABS(CBZ)                                                           016890
        IAAW=RILA+Q                                                             016900
        AAW=IAAW                                                                016910
        RILA=RILA-AAW                                                           016920
        IAAW=RILA*(10.**JK)+.49                                                 016930
        IF (IAAW.EQ.0) GO TO 2                                                  016940
        DO 1 I=1,JK                                                             016950
        NJK=I                                                                   016960
        IAAW=10.*(RILA+Q)                                                       016970
        AAW=IAAW                                                                016980
        RILA=10.*RILA-AAW                                                       016990
        IF (RILA.LE.Q) GO TO 2                                                  017000
1       CONTINUE                                                                017010
2       JK=NJK                                                                  017020
        APA=JK                                                                  017030
        CBZ=APA+1.                                                              017040
        RETURN                                                                  017050
        END                                                                     017060
```

```
      SUBROUTINE GROUP (XDISM,XDISS,YDISE,YDISV,IIE,KKV,JJN)                017070
C                                                                           017080
C     THIS SUBROUTINE IS USED IN DETERMINNING THE SIZE OF THE TRANSFER      017090
C     FUNCTION  FOR PLOTTING IT                                             017100
C                                                                           017110
      DIMENSION XDISM(1),XDISS(1),YDISE(1),YDISV(1),IIE(1)                  017120
      E=.0001                                                               017130
      LQ=0                                                                  017140
      DO 2 I=1,JJN                                                          017150
      IF (LQ.GE.I) GO TO 2                                                  017160
      LQ=I                                                                  017170
      IF ((ABS(YDISE(I)).GT.E).OR.(ABS(XDISM(I)).GT.E)) GO TO 2             017180
      KKV=KKV+1                                                             017190
      IIE(KKV)=1                                                            017200
      YDISV(KKV)=1.                                                         017210
1     IF (LQ.GE.JJN) GO TO 2                                                017220
      IF (YDISE(I).NE.YDISE(LQ+1).OR.XDISM(I).NE.XDISM(LQ+1)) GO TO 2       017230
      IIE(KKV)=2                                                            017240
      LQ=LQ+1                                                               017250
      YDISV(KKV)=YDISV(KKV)+1.                                              017260
      GO TO 1                                                               017270
2     CONTINUE                                                              017280
      LQ=0                                                                  017290
      DO 4 I=1,JJN                                                          017300
      IF (LQ.GE.I) GO TO 4                                                  017310
      LQ=I                                                                  017320
      IF ((ABS(YDISE(I)).GT.E).OR.(ABS(XDISM(I)).LT.E)) GO TO 4             017330
      KKV=KKV+1                                                             017340
      IIE(KKV)=4                                                            017350
      YDISV(KKV)=1.                                                         017360
```

```
      XDISS(KKV)=-XDISM(I)                                                    017370
3     IF (LQ.GE.JJN) GO TO 4                                                  017380
      IF (YDISE(I).NE.YDISE(LQ+1).OR.XDISM(I).NE.XDISM(LQ+1)) GO TO 4         017390
      IIE(KKV)=5                                                              017400
      LQ=LQ+1                                                                 017410
      YDISV(KKV)=YDISV(KKV)+1.                                                017420
      GO TO 3                                                                 017430
4     CONTINUE                                                                017440
      LQ=0                                                                    017450
      DO 5 I=1,JJN                                                            017460
      IF (LQ.GE.I) GO TO 5                                                    017470
      LQ=I                                                                    017480
      IF (ABS(YDISE(I)).LT.E) GO TO 5                                         017490
      KKV=KKV+1                                                               017500
      IIE(KKV)=0                                                              017510
      XDISS(KKV)=0.                                                           017520
      KKV=KKV+1                                                               017530
      LQ=LQ+1                                                                 017540
      IIE(KKV)=5                                                              017550
      XDISS(KKV)=XDISM(I)*XDISM(I)+YDISE(I)*YDISE(I)                          017560
      IF (ABS(XDISM(I)).LT.E) GO TO 5                                         017570
      IIE(KKV-1)=8                                                            017580
      IIE(KKV)=7                                                              017590
      XDISS(KKV-1)=-2.*XDISM(I)                                               017600
5     CONTINUE                                                                017610
      RETURN                                                                  017620
      END                                                                     017630
```

```
      OVERLAY(CCPOSD,1,2)                                                    017640
      PROGRAM ROOTS                                                          017650
C                                                                            017660
C     THIS PROGRAM IS USED TO FIND THE ROOTS OF INTEREST FOR A               017670
C     SPECIFIED GAIN, FOR OPTIONS 1, 2, 3, 6, AND 7                          017680
C                                                                            017690
      COMMON /MAIN012/ GANE,NOP_OT                                           017700
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME         017710
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XK,YY                                 017720
      DIMENSION XXP(50),YYP(50),ZZP(50)                                      017730
      LOGICAL NOP_OT                                                         017740
C     COMPUTE NUMERATOR POLYNOMIAL COEFFICIENTS                              017750
      NN=N                                                                   017760
      A=1.                                                                   017770
      IF(SIG.EQ.0.) A=-1.                                                    017780
      LLI=0                                                                  017790
      ZZP(1)=1.                                                              017800
      DO 4 I=1,N                                                             017810
      IF(LLI.EQ.1) GO TO 3                                                   017820
      C=0.                                                                   017830
      B=-XP(I)                                                               017840
      IF(YP(I).EQ.0.) GO TO 1                                                017850
      C=B*B+YP(I)*YP(I)                                                      017860
      B=2.*B                                                                 017870
      LLI=1                                                                  017880
1     KK=I                                                                   017890
      XXP(1)=0.                                                              017900
      XXP(2)=0.                                                              017910
      DO 2 L=1,KK                                                            017920
      XXP(L+2)=0.                                                            017930
```

```
      XXP(L)=XXP(L)+ZZP(L)                                              017940
      XXP(L+1)=XXP(L+1)+B*ZZP(L)                                        017950
      XXP(L+2)=XXP(L+2)+C*ZZP(L)                                        017960
    2 ZZP(L)=XXP(L)                                                     017970
      ZZP(I+1)=XXP(I+1)                                                 017980
      ZZP(I+2)=XXP(I+2)                                                 017990
      GO TO 4                                                           018000
    3 LLI=0                                                             018010
    4 CONTINUE                                                          018020
      YYP(1)=1.                                                         018030
      IF(M.EQ.0) GO TO 9                                                018040
C     NOW COMPUTE DENOMINATOR                                           018050
      LLI=0                                                             018060
      ZZP(1)=1.                                                         018070
      DO 8 I=1,4                                                        018080
      IF(LLI.EQ.1) GO TO 7                                              018090
      C=0.                                                              018100
      B=-XZ(I)                                                          018110
      IF(YZ(I).EQ.0.) GO TO 5                                           018120
      C=B*B+YZ(I)*YZ(I)                                                 018130
      B=2.*B                                                            018140
      LLI=1                                                             018150
    5 KK=I                                                              018160
      YYP(1)=0.                                                         018170
      YYP(2)=0.                                                         018180
      DO 6 L=1,KK                                                       018190
      YYP(L+2)=0.                                                       018200
      YYP(L)=YYP(L)+ZZP(L)                                              018210
      YYP(L+1)=YYP(L+1)+B*ZZP(L)                                        018220
      YYP(L+2)=YYP(L+2)+C*ZZP(L)                                        018230
```

```
    5     ZZP(L)=YYP(L)                                                    018240
          ZZP(I+1)=YYP(I+1)                                                018250
          ZZP(I+2)=YYP(I+2)                                                018260
          GO TO 8                                                          018270
    7     LLI=0                                                            018280
    8     CONTINUE                                                         018290
    9     LLI=N+1                                                          018300
          KK=N-M                                                           018310
          DO 10 I=1,LLI                                                    018320
          ZZP(I)=XXP(I)                                                    018330
          IF(I.LE.KK) GO TO 10                                             018340
          LI=I-KK                                                          018350
          ZZP(I)=ZZP(I)+YYP(LI)*ALPHA/GA                                   018360
   10     CONTINUE                                                         018370
          CALL SMULR (ZZP,NN,XXP,YYP)                                      018380
          WRITE(6,12)                                                      018390
          DO 11 I=1,N                                                      018400
          IF(ABS(XXP(I)).LT.E) XXP(I)=0.                                   018410
          IF(ABS(YYP(I)).LT.E) YYP(I)=0.                                   018420
          IF (NOPLOT) GO TO 11                                             018430
          VIX=YY+G*XXP(I)                                                  018440
          VIY=XX+G*YYP(I)                                                  018450
          IF ((VIX.LT..5).OR.(VIX.GT.(FX-.5)).OR.(VIY.LT.1.25).OR.(VIY.GT.(F018460
         1Y-.75))) GO TO 11                                                018470
          CALL SYMBOL (VIX,VIY,.07,2,0.0,-1)                               018480
   11     WRITE(6,13) XXP(I),YYP(I)                                        018490
          WRITE(6,14)                                                      018500
C                                                                          018510
C     FORMATS                                                              018520
C                                                                          018530
```

```
 12   FORMAT (/,6X,17HROOTS OF INTEREST)                                018540
 13   FORMAT (8X,3HX =,G13.5,9X,3HY =,G13.5)                            018550
 14   FORMAT(2X)                                                        018560
      END                                                               018570


      OVERLAY(COPOSD,1,3)                                               018580
      PROGRAM ROOT12                                                    018590
C                                                                       018600
C     TRACE ROOT LOCUS AND PRINT AND PLOT AS INDICATED BY LL CODE       018610
C                                                                       018620
       COMMON /MAIN02/OGIN,ONIG,GAG,II,LN,DPR(50)                       018630
C     DPR IS CALLED DELPR IN ROOT10                                     018640
      COMMON /MAIN012/GANE,NOPLOT,GTOL,IIV,LL,V,XAX,YAX,DEL(50),        018650
     +  XOISL(50),XOISR(50),YOISD(50),YOISU(50),XOT,YOT,CB             018660
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME    018670
      COMMON/ANGSS/A,B,ANG,PI                                           018680
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                            018690
      COMMON /ANGUU/ AA,BB,CC,DD,D,DA, JK,SANG,CA                       018700
      LOGICAL NOPLOT                                                    018710
      DATA DAN/.2/,LIE/0/,EEB/.00001/                                   018720
      JJ = JJV = 0                                                      018730
      DI = D                                                            018740
      TOPR=DPR(1)                                                       018750
      GO TO (79,79,79,76,78) LL                                         018760
C                                                                       018770
C     CODE FOR OPTIONS 4 AND 5                                          018780
C                                                                       018790
```

```
C     BEGIN ITERATION, OPTION 4                                    018800
75     WRITE (6,171) JK                                            018810
C     BEGIN ITERATION, OPTION 5                                    018820
 78    WRITE (6,173) X,Y,D,OGIN,ONIG                               018830
C                                                                  018840
C      IF LN NE 0 START TRACE AT A POLE                            018850
C                                                                  018860
       IF (LN.NE.0) GO TO 80                                       018870
       CALL ANG1                                                   018880
       A=-FY                                                       018890
       B=FX                                                        018900
       CALL BANG                                                   018910
       CALL BOX                                                    018920
       SANG=ANG                                                    018930
       GO TO 89                                                    018940
C                                                                  018950
C      START A BRANCH TRACE                                        018960
C                                                                  018970
C     BEGIN ITERATION, OPTIONS 1, 2 & 3                            018980
 79    WRITE(6,175)II,D,TOPR                                       018990
       WRITE(6,1751)OGIN,ONIG                                      019000
       JJV = 2                                                     019010
       J=1                                                         019020
       X=XP(II)                                                    019030
       Y=YP(II)                                                    019040
80     CALL GANG1                                                  019050
C                                                                  019060
C     TEST FOR MULTIPLE POLES AND PLOT                             019070
C     JJ=0, INITIALLY                                              019080
C                                                                  019090
```

```
      IF (JJ-1) 82,88,88                                                    019100
81    II=II+1                                                               019110
      GO TO 85                                                              019120
82    IJ=II+JJ                                                              019130
      IF (XP(IJ)-X) 85,83,85                                                019140
83    IF (YP(IJ)-Y) 85,84,85                                                019150
84    JJ=JJ+1                                                               019160
85    IF (JJ+II-N) 82,82,87                                                 019170
85    IF (LL-4) 87,81,81                                                    019180
87    ZP=JJ                                                                 019190
      IF(NOPLOT) GO TO 88                                                   019200
      IF (JJ.EQ.1) GO TO 88                                                 019210
      VIX=V*X+YAX                                                           019220
      VIY=V*Y+XAX                                                           019230
      IF ((VIX.LT..5).OR.(VIX.GT.(XOT-.5)).OR.(VIY.LT.1.25).OR.(VIY.GT.(019240
     1YOT-.75))) GO TO 88                                                   019250
      CALL SYMBOL (VIX,VIY,.07,11,0.0,-1)                                   019260
      CALL SYMBOL (VIX+.055,VIY+.03,.07,3H( ),0.0,3)                        019270
      CALL NUMBER (VIX+.125,VIY+.03,.07,FLOAT(JJ),0.0,-1)                   019280
      IIV=IIV+1                                                             019290
      YDISD(IIV)=VIY+.02                                                    019300
      YDISU(IIV)=VIY+.11                                                    019310
      XDISL(IIV)=VIX+.05                                                    019320
      XDISR(IIV)=VIX+.23                                                    019330
88    Z=JJ                                                                  019340
      JJ=JJ-1                                                               019350
C     ADJUST PHASE ANGLE AT REPEATED POLE                                   019360
      ANG=(GANG+2.*PI*Z)/ZP                                                 019370
      LN=1                                                                  019380
      GAIN=0.                                                               019390
```

```
      CALL BOX                                                          019400
      SANG=ANG                                                          019410
      CA=SANG                                                           019420
88    DA=D                                                              019430
      DS=D                                                              019440
      L=1                                                               019450
      JL=1                                                              019460
      K=1                                                               019470
90    XX=X                                                              019480
      YY=Y                                                              019490
      GEN=GAIN                                                          019500
      A=X                                                               019510
      B=Y                                                               019520
      WNN=SQRT(A*A+B*B)                                                 019530
      CALL BANG                                                         019540
      ZE=ABS(COS(ANG))                                                  019550
      DE=1.02*DI                                                        019560
      IF (LL.EQ.4) GO TO 91                                             019570
C                                                                       019580
C     CHECK FOR AT AN OUTSIDE BOUNDARY                                  019590
C                                                                       019600
      IF (XX.GT.AA.OR.YY.GT.BB.OR.XX.LT.CC.OR.YY.LT.DD) GO TO 162       019610
C                                                                       019620
C    TEST TO SEE IF POINT SHOULD BE PRINTED OR SKIPPED,                 019630
C    PRINT EVENLY SPACED POINTS ACCORDING TO PRINTING STEP SIZE         019640
C                                                                       019650
 91   IF ((LN.NE.0).OR.(LL.GE.4)) GO TO 92                              019660
      SDPR=SDPR+D                                                       019670
      IF(SDPR .LT. TDPR) GO TO 93                                       019680
C     WRITE LOCUS POINT AND RESET SPACING COUNTER                       019690
```

```
 92   WRITE (6,185) XX,YY,WIN,GAIN,ZE,LN                                    019700
      SDPR=TDPR*.000001                                                     019710
C                                                                           019720
C     SKIP PLOT IF NOPLOT                                                   019730
C                                                                           019740
 93   IF(NOPLOT) GO TO 99                                                   019750
C                                                                           019760
C     SET UP TO PLOT POINT                                                  019770
C                                                                           019780
      VIX=V*X+XAX                                                           019790
      VIY=V*Y+XAX                                                           019800
C     ONLY PLOT AN X FOR A POLE                                             019810
      IF (LN.EQ.1) CALL SYMBOL(VIX,VIY,.07,4,0,-1)                          019820
C                                                                           019830
C    CHECK BLOCKED OUT AREAS OF PLOT                                        019840
C                                                                           019850
      IF(JJV .EQ. 3) CALL PLOT(VIX,VIY,3)                                   019860
      JJV=3                                                                 019870
      DO 94 I=1,IIV                                                         019880
      IF ((VIX.LT.XDISR(I)).AND.(VIX.GT.XDISL(I)).AND.(VIY.LT.YDISU(I)).019890
     1AND.(VIY.GT.YDISD(I))) GO TO 96                                       019900
 94    CONTINUE                                                             019910
      JJV=2                                                                 019920
C    PLOT LOCUS POINT - PEN UP IF IN BLOCKED OUT AREA                       019930
 96   CALL PLOT(VIX,VIY,JJV)                                                019940
 99   IF(M .EQ. 0) GO TO 110                                                019950
C                                                                           019960
C     TEST POINT TO SEE IF IT IS NEAR A ZERO                                019970
C                                                                           019980
      DO 109 I=1,M                                                          019990
```

```
      IF (ABS(XX-XZ(I))-DE) 101,101,109                                 020000
101   IF (ABS(YY-YZ(I))-DE) 102,102,109                                 020010
C                                                                       020020
C     IT IS CLOSE - SEE IF THERE IS A POLE CLOSER                       020030
C                                                                       020040
102   DO 107 IJ=1,N                                                    020050
      IF (ABS(XZ(I)-XP(IJ))-2.*DE) 103,103,107                          020060
103   IF (ABS(YZ(I)-YP(IJ))-2.*DE) 104,104,107                          020070
104   A=XZ(I)-XP(IJ)                                                    020080
      B=YZ(I)-YP(IJ)                                                    020090
      DA=.2*SQRT(A*A+B*B)                                               020100
      IF (ABS(XX-XZ(I))-DA) 105,105,106                                 020110
105   IF (ABS(YY-YZ(I))-DA) 108,105,106                                 020120
106   IF (DA.LT.(.0001)) GO TO 108                                      020130
      D=DA                                                              020140
      GO TO 110                                                         020150
107   CONTINUE                                                          020160
C                                                                       020170
C     A ZERO IS REACHED, WRITE AND PLOT IT                              020180
C                                                                       020190
108   GAIN=0.                                                           020200
      LN=2                                                              020210
      A=XZ(I)                                                           020220
      B=YZ(I)                                                           020230
      WNN=SQRT(A*A+B*B)                                                 020240
      CALL BANG                                                         020250
      ZE=ABS(COS(ANG))                                                  020260
      WRITE (6,185) XZ(I),YZ(I),WNN,GAIN,ZE,LN                          020270
      IF (NOPLOT) GO TO 163                                             020280
      VIX=V*XZ(I)+YAX                                                   020290
```

```
      VIY=V*YZ(I)+XAX                                                  020300
      CALL SYMBOL (VIX,VIY,.07,1,0.0,-2)                               020310
C                                                                      020320
C     NEXT BRANCH OR LOCUS                                             020330
C                                                                      020340
      GO TO 163                                                        020350
109   CONTINUE                                                         020360
C                                                                      020370
C     NOT A ZERO - GO ON                                               020380
C                                                                      020390
      DA=DI                                                            020400
      D=DS                                                             020410
C     J = 4 FOR A BREAK POINT                                          020420
110   GO TO (111,111,116,153,111), J                                   020430
C                                                                      020440
C     SEARCH ALONG REAL AXIS                                           020450
C                                                                      020460
111   IF (ABS(YY)-E) 112,112,121                                       020470
112   CB=1.                                                            020480
      CAA=0.                                                           020490
      CA=0.                                                            020500
      IF (ABS(SANG)-.3) 115,115,113                                    020510
113   IF (ABS(SANG+PI)-.3) 115,115,114                                 020520
114   IF (ABS(SANG-PI)-.3) 115,115,121                                 020530
115   CB=-1.                                                           020540
      CAA=-PI                                                          020550
      CA=-PI                                                           020560
116   X=XX+CB*D                                                        020570
      YY=0.                                                            020580
      Y=0.                                                             020590
```

```
      J=3                                                                020600
      CALL ANG1                                                          020610
      IF (J.EQ.3) GO TO 117                                              020620
      J=3                                                                020630
      GO TO 120                                                          020640
117   IF (ABS(GANG).GT.(.5)) GO TO 120                                   020650
      SANG=0.                                                            020660
      IF (FY.GT.0.) SANG=-PI                                             020670
      IF (ABS(FY).EQ.0.) GO TO 119                                       020680
      IF ((ABS(ABS(SANG)-ABS(CA))).GT.2.5) GO TO 118                     020690
      IF (LIE.NE.1) GO TO 1170                                           020700
      X=XX                                                               020710
1170  LIE=0                                                              020720
      IF (K.GE.5.OR.K.EQ.1) GO TO 132                                    020730
      K=K+1                                                              020740
      XY=XX                                                              020750
      XX=X                                                               020760
      GO TO 115                                                          020770
118   IF (LIE.EQ.1) GO TO 152                                            020780
      K=2                                                                020790
      D=D/4.                                                             020800
      IF (DA/255.-D) 116,115,152                                         020810
119   XX=X                                                               020820
      LIE=1                                                              020830
      GO TO 115                                                          020840
C                                                                        020850
C     REDUCE STEPPING DISTANCE AND TRY AGAIN                             020860
C                                                                        020870
120   D=D/4.                                                             020880
      IF (D.LT.(DA/260.)) GO TO 150                                      020890
```

```
      IF (J.EQ.3) GO TO 115                                          020900
C                                                                    020910
C     SEARCH COMPLEX POINTS                                          020920
C                                                                    020930
121   J=1                                                            020940
      CA=SANG                                                        020950
C                                                                    020960
C     MOVE TO A NEW POINT                                            020970
C                                                                    020980
122   X=XX+D*COS(SANG)                                               020990
      Y=YY+D*SIN(SANG)                                               021000
      CALL ANG1                                                      021010
      IF (J.EQ.3) GO TO 120                                          021020
      IF (ABS(GANG)-.00001) 124,124,123                              021030
C                                                                    021040
C     IF OFF LOCUS, TRY TO IMPROVE                                   021050
C                                                                    021060
 123  IF ((L.EQ.2).AND.(JL.GE.3)) GO TO 130                          021070
      ANG=SANG-GANG/(FY*D*COS(SANG)-FX*D*SIN(SANG))                  021080
      CALL BOX                                                       021090
      SANG=ANG                                                       021100
      JL=JL+1                                                        021110
      IF (JL-20) 122,122,125                                         021120
124   GO TO (126,130,126), L                                         021130
125   JL=1                                                           021140
      X=XX+D*COS(SANG)                                               021150
      Y=YY+D*SIN(SANG)                                               021160
      IF (ABS(GANG)-.0005) 129,129,161                               021170
C                                                                    021180
C     PERTURB POINT BY EEB IN EACH DIRECTION                         021190
```

```
C     IF ERROR ANGLE CHANGES IN EITHER DIRECTION WE ARE WITHIN          021200
C     .00001 OF THE LOCUS                                               021210
C                                                                       021220
125   R=SQRT(D*D+EEB*EEB)                                               021230
      XXX=X                                                             021240
      YYY=Y                                                             021250
      ED=EEB/D                                                          021260
      GAG=SANG+ATAN(ED)                                                 021270
      X=XX+R*COS(GAG)                                                   021280
      Y=YY+R*SIN(GAG)                                                   021290
      CB=GANG                                                           021300
      CALL GANG1                                                        021310
C                                                                       021320
C     MY MODIFICATION ALLOWS FOR 1E-10 ERROR IN THE ANGULAR COMPUTATION 021330
C                                                                       021340
      IF(GANG*CB-1.E-20) 128,128,127                                    021350
127   GAG=SANG-ATAN(ED)                                                 021360
      X=XX+R*COS(GAG)                                                   021370
      Y=YY+R*SIN(GAG)                                                   021380
      CALL GANG1                                                        021390
      IF(GANG*CB-1.E-20 .LE. 0) GO TO 128                               021400
      GANG=CB                                                           021410
      GO TO 123                                                         021420
C                                                                       021430
C     WE ARE ON LOCUS - PROCEED                                         021440
C                                                                       021450
128   JL=1                                                              021460
      X=XXX                                                             021470
      Y=YYY                                                             021480
129   CAA=CA                                                            021490
```

```
      CA=SANG                                                                021500
C                                                                            021510
C     COMPUTE SLOPE AT THE NEW POINT                                         021520
C     IF THE CHANGE IN SLOPE FROM THE OLD POINT (DANG IS THE CHANGE)         021530
C     IS GREATER THAN PI/5, ASSUME THAT WE HAVE PASSED A BREAK POINT         021540
C                                                                            021550
130   A=-FY                                                                  021560
      B=FX                                                                   021570
      CAPOT=SANG                                                             021580
      CALL BANG                                                              021590
      CALL BOX                                                               021600
      SANG=ANG                                                               021610
      IF (L.EQ.2) GO TO 155                                                  021620
      DANG=ABS(SANG-CA)                                                      021630
      EANG=ABS(CAA-CA)                                                       021640
      IF (EANG.GT.1.4) GO TO 149                                             021650
      IF ((EANG.GT..3).AND.(K.GE.2)) GO TO 149                               021660
 131  IF(DANG .GT. .6) GO TO 150                                             021670
C     ALLOW A SMALLER CHANGE IF WE THINK WE ARE NEAR A BREAK POINT           021680
      IF ((DANG.GT.DAN).AND.(K.GE.2)) GO TO 150                              021690
132   XY=XX                                                                  021700
      YX=YY                                                                  021710
      XX=X                                                                   021720
      YY=Y                                                                   021730
      GO TO (134,147,147,147,133), K                                         021740
133   LIE=0                                                                  021750
      K=1                                                                    021760
      D=DA                                                                   021770
134   IF (XY*XX) 136,135,140                                                 021780
135   IF (XY) 136,140,136                                                    021790
```

```
135   LN=4                                                                    021800
      XX=XY                                                                   021810
      YY=YX                                                                   021820
      D=D/4.                                                                  021830
      IF (DA/256.-D) 137,137,138                                              021840
137   L=3                                                                     021850
      IF (J-3) 122,139,122                                                    021860
138   Y=YX-(YX-Y)*XY/(XY-X)                                                   021870
139   X=0.                                                                    021880
      D=DA                                                                    021890
      CALL ANG1                                                               021900
      GO TO 89                                                                021910
140   IF (L-3) 141,122,141                                                    021920
141   D=DA                                                                    021930
      LN=0                                                                    021940
      GO TO (143,143,143,142,143), LL                                         021950
C                                                                             021960
C     DECREMENT AND TEST POINT COUNTER FOR OPTION 4                           021970
C                                                                             021980
142   JK=JK-1                                                                 021990
      IF (JK) 163,163,143                                                     022000
C                                                                             022010
C     DETERMINE WHETHER SOLUTION IS AT GAIN OF INTEREST                       022020
C      THEN GO TO PLOT IT                                                     022030
C                                                                             022040
143   IF (GAIN.LT.(GANE-GTOL)) GO TO 90                                       022050
      IF (GTOL.EQ.0.) GO TO 146                                               022060
      IF (((GEN-GANE).LT.0.).AND.((GAIN-GANE).GE.0.)) GO TO 144               022070
      IF (GAIN.GT.(GANE+GTOL)) GO TO 146                                      022080
      GO TO 145                                                               022090
```

```
 144   IF (NOPLOT) GO TO 145                                          022100
       VIY=Y*V+XAX                                                    022110
       VIX=V*X+YAX                                                    022120
       IF(TIME.NE.0.)CALL SYMBOL (VIX,VIY,.07,2,0.0,-2)               022130
C     REDUCE STEP SIZE AROUND GAIN OF INTEREST                        022140
145    DS=DI/5.                                                       022150
       D=DS                                                           022160
       LN=5                                                           022170
       GO TO 90                                                       022180
146    DS=DI                                                          022190
       D=DS                                                           022200
       GO TO 90                                                       022210
C                                                                     022220
C      SNEAK UP ON A BREAK POINT - INITIAL ENTRY AT 150               022230
C                                                                     022240
147    K=K+1                                                          022250
148    IF (J-3) 122,116,122                                           022260
149    IF ((ABS(CAA).GT.2.5).AND.(ABS(CA).GT.2.5)) GO TO 131          022270
       CA=CAA                                                         022280
       GO TO 151                                                      022290
150    IF ((ABS(CA).GT.2.8).AND.(ABS(SANG).GT.2.8)) GO TO 132         022300
151    SANG=CA                                                        022310
       K=2                                                            022320
       D=D/4.                                                         022330
       IF (D.LT.E) GO TO 152                                          022340
       IF (DA/256.-D) 148,148,152                                     022350
C                                                                     022360
C      COMPUTE EXACT BREAK POINT AND CHECK TO SEE IF ON REAL AXIS     022370
C                                                                     022380
152    CALL BREAK                                                     022390
```

```
      IF (J.EQ.3) YY=0.                                                  022400
      J=4                                                                022410
      IF (ABS(YY).LT..001) YY=0.                                         022420
      CA=CAA                                                             022430
      LIE=0                                                              022440
      X=XX                                                               022450
      Y=YY                                                               022460
      LN=3                                                               022470
      CALL ANG1                                                          022480
      J=4                                                                022490
      GO TO 90                                                           022500
C                                                                        022510
C     APPEARS TO BE ROUTINE TO LEAVE BREAK POINT                         022520
C                                                                        022530
153   K=2                                                                022540
      D=DA/100.                                                          022550
154   DO 158 JM=1,6                                                      022560
      J=1                                                                022570
      ANG=CA-(11-JM )*PI/12.                                             022580
      CALL BOX                                                           022590
      PHI=ANG                                                            022600
      SANG=ANG                                                           022610
      L=2                                                                022620
      GO TO 122                                                          022630
C                                                                        022640
C     CHECK PHASE CRITERION                                              022650
C                                                                        022660
155   JL=1                                                               022670
      IF (ABS(CAPOT-PHI).LT.(.27)) GO TO 156                             022680
      IF ((ABS(CAPOT).LT.(2.3)).OR.(ABS(PHI).LT.(2.9))) GO TO 158        022690
```

```
      IF ((ABS(ABS(CAPOT)-ABS(PHI))).GT.(.27)) GO TO 158                022700
156   IF (ABS(CAPOT-SANG).LT.(.5)) GO TO 157                            022710
      IF ((ABS(SANG).LT.(2.9)).OR.(ABS(CAPOT).LT.(2.9))) GO TO 158      022720
      IF ((ABS(ABS(CAPOT)-ABS(SANG))).GT.(.5)) GO TO 158                022730
157   D=DA                                                              022740
      SANG=CAPOT                                                        022750
      CA=SANG                                                           022760
      CAA=CA                                                            022770
      L=1                                                               022780
      K=1                                                               022790
      GO TO 111                                                         022800
158   CONTINUE                                                          022810
      IF (D.GE.(DA/11.)) GO TO 159                                      022820
      D=DA/10.                                                          022830
      GO TO 154                                                         022840
C                                                                       022850
C     TROUBLE - PRINT MESSAGE AND TERMINATE BRANCH                      022860
C                                                                       022870
159   WRITE (6,188)                                                     022880
      GO TO 163                                                         022890
160   WRITE (6,182)                                                     022900
      GO TO 163                                                         022910
161   WRITE (6,187)                                                     022920
      GO TO 163                                                         022930
C                                                                       022940
C     WRITE BOUNDARY MESSAGES                                           022950
C                                                                       022960
C    POLE OUTSIDE RANGE                                                 022970
162   IF(LN .NE. 1) GO TO 167                                           022980
      WRITE(6,183)X,Y                                                   022990
```

```
      GO TO 163                                                          023000
C     LOCUS CROSSED BOUNDARY                                             023010
 157  WRITE(6,185)XX,YY,WNN,GAIN,ZE,LN                                   023020
      WRITE(6,186)                                                       023030
C                                                                        023040
C     FINISHED A BRANCH - ACTION DEPENDS ON OPTION                       023050
C                                                                        023060
163   D=DI                                                               023070
      SDPR=0.                                                            023080
      GO TO (165,154,165,200,200) LL                                     023090
 154  IF (II.GT.N) GO TO 165                                             023100
      D=DEL(II+1)                                                        023110
      TDPR=DPR(II+1)                                                     023120
      DI=D                                                               023130
165   IF (JK.EQ.5) GO TO 166                                             023140
      II=II+1                                                            023150
      IF (II.LE.N) GO TO 79                                              023160
C                                                                        023170
C     SEARCH BOUNDARIES                                                  023180
C                                                                        023190
166   CALL SEEK                                                          023200
C                                                                        023210
C     LOCATE BRANCH STARTING OUTSIDE RANGE                               023220
C                                                                        023230
      IF(JK .EQ. 1) GO TO 200                                            023240
      WRITE(6,1730)X,Y,D,TDPR                                            023250
      WRITE(6,1751)CGIN,CNIG                                             023260
      IF(NOPLOT) GO TO 111                                               023270
      VIX=V*X+YAX                                                        023280
      VIY=V*Y+XAX                                                        023290
```

```
      CALL PLOT(VIX,VIY,3)                                              023300
      JJV = 2                                                           023310
      GO TO 111                                                         023320
C                                                                       023330
C     EXIT                                                              023340
C                                                                       023350
 200  CONTINUE                                                          023360
C                                                                       023370
C     FORMATS                                                           023380
C                                                                       023390
 171  FORMAT (/,6X,23HNUMBER OF POINTS (JK) =,I5)                       023400
 173  FORMAT (///,9X,25HSUB-BRANCH STARTING AT X =,G13.5,4X,3HY =,G13.5,023410
     1//,6X,45HCALCULATION STEP SIZE = PRINTING STEP SIZE = ,G10.4,//,3X023420
     2,10HLOCUS REAL,5X,10HLOCUS IMAG,3X,14HDIST TO ORIGIN,1X,2A6,4X,4HZ023430
     3ETA,4X,2HOD)                                                      023440
1730 FORMAT (///,9X,25HSUB-BRANCH STARTING AT X =,G13.5,4X,3HY =,G13.5,023450
     1//,6X,24HCALCULATION STEP SIZE = ,G10.4,/,6X,24HPRINTING STEP SIZE023460
     2    = ,G10.4)                                                    023470
 175  FORMAT (///,23X,14HBRANCH NUMBER ,I3,//,6X,24HCALCULATION STEP SIZ023480
     1E = ,G10.4,/,6X,24HPRINTING STEP SIZE    = ,G10.4)               023490
1751 FORMAT (/,3X,10HLOCUS REAL,5X,10HLOCUS IMAG,3X,14HDIST TO ORIGIN,1023500
     1X,2A6,4X,4HZETA,4X,2HOD)                                          023510
182   FORMAT (10X21HTROUBLES NEAR A POLE./10X32HREDUCING THE STEP SIZE M023520
     1AY HELP.)                                                         023530
 183  FORMAT (10X,15HTHE POLE AT X =,G13.5,8H AND Y =,G13.5,/,10X,24HIS 023540
     1OUTSIDE THE BOUNDARY.)                                            023550
 185  FORMAT (1X,3(G14.8,1X),G12.6,1X,F9.5,2X,I2)                       023560
186   FORMAT (10X,8HBOUNDARY)                                           023570
187   FORMAT (10X,59HTROUBLE IN FINDING THE NEXT POINT TO WITHIN 10E-5 A023580
     1CCURACY./10X,32HREDUCING THE STEP SIZE MAY HELP.)                 023590
```

```
188   FORMAT (10X,30HTROUBLE LEAVING A BREAK POINT./10X49HUSE OPTION 5 T023600
     10 GET A CONTINUATION OF THIS BRANCH)                              023610
      END                                                               023620


      SUBROUTINE BREAK                                                  023630
C                                                                       023640
C     ZERO IN ON A BREAK POINT                                          023650
C                                                                       023660
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME    023670
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                            023680
      IIM=0                                                             023690
1     IF (IIM.GT.10) GO TO 4                                            023700
      FX=0.                                                             023710
      FY=-TIME                                                          023720
      FXX=0.                                                            023730
      FXY=0.                                                            023740
      DO 3 I=1,N                                                        023750
      G=(XX-XP(I))*(XX-XP(I))+(YY-YP(I))*(YY-YP(I))                     023760
      IF( G .EQ. 0) GO TO 2                                             023770
      FX=FX+(YY-YP(I))/G                                                023780
      FY=FY-(XX-XP(I))/G                                                023790
      FXX=FXX-2.*(XX-XP(I))*(YY-YP(I))/(G*G)                            023800
      FXY=FXY+((XX-XP(I))*(XX-XP(I))-(YY-YP(I))*(YY-YP(I)))/(G*G)       023810
2     IF(M .LT. I) GO TO 3                                              023820
      G=(XX-XZ(I))*(XX-XZ(I))+(YY-YZ(I))*(YY-YZ(I))                     023830
      IF(G .EQ. 0) GO TO 3                                              023840
      FX=FX-(YY-YZ(I))/G                                                023850
```

```
      FY=FY+(XX-XZ(I))/G                                                    023860
      FXX=FXX+2.*(XX-XZ(I))*(YY-YZ(I))/(G*G)                                023870
      FXY=FXY-((XX-XZ(I))*(XX-XZ(I))-(YY-YZ(I))*(YY-YZ(I)))/(G*G)           023880
3     CONTINUE                                                              023890
      IF( ABS(FXX) + ABS(FXY) .LT. 1.E-10) GO TO 4                          023900
      XX=XX-(FX*FXX+FY*FXY)/(FXX*FXX+FXY*FXY)                               023910
      YY=YY+(FY*FXX-FX*FXY)/(FXX*FXX+FXY*FXY)                               023920
      IIM=IIM+1                                                             023930
      IF ((ABS(FY).GT.0.).OR.(ABS(FX).GT.0.)) GO TO 1                       023940
4     RETURN                                                                023950
      END                                                                   023960


      SUBROUTINE SEEK                                                       023970
C                                                                           023980
C     THIS SUBROUTINE SEARCHES THE BOUNDARIES FOR ANY LOCUS BRANCHES        023990
C     WHICH RE-ENTER THE REGION OF CALCULATION AFTER LEAVING IT OR          024000
C     FOR ANY LOCUS BRANCHES WHICH START OUTSIDE OF THE REGION OF           024010
C     CALCULATION AND THEN ENTER IT                                         024020
C                                                                           024030
      COMMON/ANGRR/XP(50),YP(50),XZ(50),YZ(50),N,M,X,Y,SIG,GANG,TIME        024040
      COMMON/ANGSS/A,B,ANG,PI                                               024050
      COMMON/ANGTT/FX,FY,GAIN,GA,G,J,E,XX,YY                                024060
      COMMON /ANGUU/ AA,BB,CC,DD,D,DA, JK,SANG,CA                           024070
      LOGICAL LOOP                                                          024080
C                                                                           024090
C     ENTRY POINT (JK=5 IS A REENTRY)                                       024100
C                                                                           024110
```

```
      DA = D                                                          024120
      LOOP = .FALSE.                                                  024130
      IF(JK .EQ. 5) GO TO 18                                          024140
      ISIDE = 1                                                       024150
      AX = 1.                                                         024150
      AY = 0                                                          024170
 1    Y=DD                                                            024180
      X=CC                                                            024190
C                                                                     024200
C     LOOP CONTROL AND HUNT LOOP                                      024210
C                                                                     024220
 2    CALL GANG1                                                      024230
 3    XA=X                                                            024240
      YA=Y                                                            024250
      CG = GANG                                                       024260
 4    X=XA+D*AX                                                       024270
      Y=YA+D*AY                                                       024280
      CALL GANG1                                                      024290
      IF(GANG*CG .LE. 0) GO TO 11                                     024300
      IF(LOOP) GO TO 3                                                024310
      GO TO (5,5,6,6)ISIDE                                            024320
 5    IF(X .GT. AA) GO TO (7,8)ISIDE                                  024330
      GO TO 3                                                         024340
 6    IF(Y .GT. BB) GO TO (9,9,9,10)ISIDE                             024350
      GO TO 3                                                         024360
 7    Y=BB                                                            024370
      X=CC                                                            024380
      ISIDE=2                                                         024390
      GO TO 2                                                         024400
 8    AX=0                                                            024410
```

```
      AY=1.                                                          024420
      ISIDE=3                                                        024430
      GO TO 1                                                        024440
 9    X=AA                                                           024450
      Y=DD                                                           024460
      ISIDE=4                                                        024470
      GO TO 2                                                        024480
C                                                                    024490
C     ALL DONE SEARCH - JK=1 IS FLAG TO CALLER                       024500
C                                                                    024510
 10   JK=1                                                           024520
      RETURN                                                         024530
C                                                                    024540
C     A BRANCH IS CROSSED                                            024550
C                                                                    024560
 11   IF(ABS(GANG) .GT. PI*.5) GO TO 3                               024570
      CALL ANG1                                                      024580
      J = 1                                                          024590
      A=-FY                                                          024600
      B=FX                                                           024610
      CALL BANG                                                      024620
      CALL BOX                                                       024630
      SANG=ANG                                                       024640
      GO TO (12,13,14,15)ISIDE                                       024650
 12   IF(SANG)3,3,16                                                 024660
 13   IF(SANG)15,3,3                                                 024670
 14   IF(ABS(SANG)-PI*.5)15,3,3                                      024680
 15   IF(ABS(SANG)-PI*.5)3,3,16                                      024690
C                                                                    024700
C     HAVE AN ENTRY BRANCH - REFINE IF NECESSARY                     024710
```

```
C                                                                       024720
 15   IF(GANG .EQ. 0) GO TO 17                                          024730
      LOOP = .TRUE.                                                     024740
      D=D*.25                                                           024750
      IF(D .GT. DA/1024.) GO TO 4                                       024760
C                                                                       024770
C     HAVE FOUND A GOOD ENTRY POINT - USE IT                            024780
C                                                                       024790
 17   YA=Y                                                              024800
      XA=X                                                              024810
      D=DA                                                              024820
      JK=5                                                              024830
      CG=GANG                                                           024840
      CA=SANG                                                           024850
      XX=X                                                              024860
      YY=Y                                                              024870
      RETURN                                                            024880
C                                                                       024890
C     REENTRY - MOVE A LITTLE FROM THE OLD POINT AND RESTART            024900
C                                                                       024910
 18   X=XA+D*AX*.0001                                                   024920
      Y=YA+D*AY*.0001                                                   024930
      GO TO 2                                                           024940
      END                                                               024950


      OVERLAY(COPOSD,2,0)                                               024960
      PROGRAM FREDR                                                     024970
```

```
      COMMON/COM/X(502),Y(502),Z(502),XDB(502),ICOUNT,TTY,A(21),B(21),   024980
     1 E(21),F(21),C(40),CTEMP(40),KORD(40),D(21)                        024990
      COMMON /CONVERT/ RAD2DEG,DEG2RAD                                   025000
      LOGICAL TTY,KFLAG                                                  025010
      DATA RAD2DEG/57.2957795/,DEG2RAD/.01745329252/                     025020
      PRINT 400                                                          025030
  400 FORMAT(//" ================================================025040
     +================"//" ***************START OF PROGRAM 'FREQR'*****025050
     +*********")                                                        025060
      IFRSEQ=2                                                           025070
      TTY = .TRUE.                                                       025080
      LPLOT=0                                                            025090
      PRINT 2050                                                         025100
 2050 FORMAT(//"  DO YOU WANT A LIST OF OPTIONS AVAILABLE FOR 'FREQR'?"/025110
     +" TYPE 'YES' OR 'NO'."/)                                           025120
 2051 READ 2052,KI                                                       025130
 2052 FORMAT(A10)                                                        025140
      IF(KI.EQ.3HYES) GO TO 2053                                         025150
      IF(KI.EQ.2HNO) GO TO 1                                             025160
      PRINT 2055                                                         025170
 2055 FORMAT("    SORRY,TRY AGAIN...TYPE 'YES' OR 'NO'.")               025180
      GO TO 2051                                                         025190
 2053 PRINT 2054                                                         025200
 2054 FORMAT(/" ***************************************************     025210
     +"/" THE FOLLOWING OPTIONS ARE AVAILABLE FOR PROGRAM 'FREQR':"//    025220
     +"      1...TO BEGIN A NEW PROBLEM. INPUT OF G(JW) OR G(Z)."/       025230
     +"     -1...FOLLOWED BY W (OMEGA); CALCULATES G(JW) USING FUNCTION"/025240
     +"          ALREADY READ INTO MEMORY AFTER OPTION 1."/             025250
     +"     -2...CLOSED-LOOP RESPONSE, C(JW)/R(JW)."/                   025260
     +"      3...PRODUCES TTY PLOT OF OPTIONS -1 OR -2, OR BOTH."/      025270
```

```
     +"    -3...PRODUCES CALCOMP PLOT OF BODE DIAGRAM (FREQ RESPONSE)."/025280
     +"     4...TO READ IN A H(JW), FEEDBACK TRANSFER FUNCTION."/       025290
     +"     5...ADDS A TRANSPORT LAG TERM (E(EXP JWT)) TO G(JW)."/      025300
     +"     6...INPUT NEW FORWARD GAIN CONSTANT ONLY."/                 025310
     +"    -6...INPUT NEW FEEDBACK GAIN CONSTANT ONLY."/                025320
     +"     7...ALLOWS AUTOMATIC DISPOSE BODE PLOTS TO BLDG 640, 'BB'."/025330
     +"     9...TO SEE PRESENT G(S), H(S), OR G(Z) FUNCTIONS."/         025340
     +"   -11...SAME AS -1, EXCEPT G(Z) IS CALCULATED."/                025350
     +"   -12...SAME AS -2, EXCEPT CLOSED-LOOP Z-DOMAIN IS CALCULATED."/025360
     +"    15...TO SET A SAMPLE PERIOD FOR THE SAMPLED DATA SYSTEM."/   025370
     +"     0...CAUSES THE PROGRAM TO STOP."/                           025380
     +" ***************************************************")          025390
1     PRINT 405                                                        025400
405   FORMAT(" TYPE OPTION NUMBER -- ")                                025410
      READ *,FIND                                                      025420
4     IND = FIND                                                       025430
      IF(IND .EQ. -1 .OR. IND .EQ. -2) GO TO 50                        025440
      MF = 0                                                           025450
      IF (IND.EQ.0) GO TO 700                                          025460
      IF(IND .EQ. 1) GO TO 8                                           025470
      IF(IND .EQ. 4) GO TO 10                                          025480
      IF(IND .EQ. 3) GO TO 200                                         025490
      IF (IND.EQ.-3) GO TO 300                                         025500
      IF(IND .EQ. 5) GO TO 20                                          025510
      IF(IND.EQ.6)GO TO 500                                            025520
      IF(IND.EQ.-6) GO TO 550                                          025530
      IF(TTY) GO TO 1                                                  025540
      PRINT 405                                                        025550
406   FORMAT("0ERROR IN OPTION.  RUN TERMINATED.")                     025560
      STOP                                                             025570
```

```
C                                                                       025580
C                                                                       025590
C     READ NUMERATOR AND DENOMINATOR OF G(S)                            025600
C                                                                       025610
C                                                                       025620
 8    KFLAG = .TRUE.                                                    025630
      IFEED = 1H                                                        025640
      PRINT 402                                                         025650
 402  FORMAT("1  FORWARD TRANSFER FUNCTION -- G(S)")                    025660
      CALL POLYRD(A,N,B,MD,OGN)                                         025670
      TAUD = 0.0                                                        025680
      ICOUNT = 0                                                        025690
      GO TO 1                                                           025700
 500  CONTINUE                                                          025710
      ICOUNT =0                                                         025720
      IF (TTY) PRINT 501                                                025730
 501  FORMAT("0TYPE NEW GAIN CONSTANT K-- ")                            025740
      READ *,GNW                                                        025750
      PRINT 503,GNW                                                     025760
 503  FORMAT("1    NEW GAIN CONSTANT = ",G15.7)                         025770
      GFACT=GNW/OGN                                                     025780
      OGN=GNW                                                           025790
      NN=N+1                                                            025800
      DO 510 II=1,NN                                                    025810
 510  A(II)=A(II)*GFACT                                                 025820
      GO TO 1                                                           025830
 550  KFLAG=.FALSE.                                                     025840
      ICOUNT =0                                                         025850
      IFEED=1HH                                                         025860
      IF(TTY) PRINT 551                                                 025870
```

```
  551 FORMAT("0TYPE NEW FEEDBACK GAIN CONSTANT K-- ")                    025880
      READ *,FBNG                                                        025890
      PRINT 553,FBNG                                                     025900
  553 FORMAT("1     NEW FEEDBACK GAIN CONSTANT =",G15.7)                 025910
      IF(FBNG.EQ.0.) GO TO 558                                           025920
      FBFACT=FBNG/FBOG                                                   025930
      FBOG=FBNG                                                          025940
      NE1=NE+1                                                           025950
      DO 555 II=1,NE1                                                    025960
  555 E(II)=E(II)*FBFACT                                                 025970
      GO TO 1                                                            025980
  558 KFLAG=.TRUE.                                                       025990
      IFEED=1H                                                           026000
      PRINT 559                                                          026010
  559 FORMAT(*1      SYSTEM NOW UNITY FEEDBACK, H=1  *)                  026020
      NE=0                                                               026030
      NF=0                                                               026040
      E(1)=1.0                                                           026050
      F(1)=1.0                                                           026060
      GO TO 1                                                            026070
C                                                                        026080
C                                                                        026090
C     READ NUMERATOR AND DENOMINATOR OF FEEDBACK TRANSFER FUNCTION       026100
C                                                                        026110
C                                                                        026120
 10   KFLAG = .FALSE.                                                    026130
      ICOUNT =0                                                          026140
      IFEED = 1HH                                                        026150
      PRINT 403                                                          026160
 403  FORMAT("-  FEEDBACK TRANSFER FUNCTION -- H(S)")                    026170
```

```
      CALL POLYRD(E,NE,F,NF,FBOG)                                       026180
      GO TO 1                                                           026190
C                                                                       026200
C                                                                       026210
C     READ TRANSPORT DELAY                                              026220
C                                                                       026230
C                                                                       026240
 20   IF(TTY) PRINT 415                                                 026250
 415  FORMAT("0TYPE TRANSPORT DELAY - TAJ -- ")                         026260
      ICOUNT =0                                                         026270
      READ *,TAJD                                                       026280
      PRINT 416,TAUD                                                    026290
 416  FORMAT("0TRANSPORT DELAY =",F15.5)                                026300
      TAUD = TAJD * RAD2DEG                                             026310
      GO TO 1                                                           026320
C                                                                       026330
C                                                                       026340
C     REQUEST PARAMETERS AND THEN COMPUTE AND PRINT RESPONSE VALUES     026350
C                                                                       026360
C                                                                       026370
  50  CONTINUE                                                          026380
      IF(TTY) PRINT 450                                                 026390
 450  FORMAT(" TO RESTART PLOT ARRAY STORAGE, ENTER A '1'."/)           026400
      IF(TTY) READ *,IFRSEQ                                             026410
      IF(IFRSEQ.EQ.1) ICOUNT =0                                         026420
      IF(TTY) PRINT 417                                                 026430
 417  FORMAT(" TYPE INITIAL, INCREMENTAL, AND FINAL OMEGA."/)           026440
      READ *,WO,DELTW,WMAX                                              026450
      W = WO                                                            026460
      IND = -IND                                                        026470
```

```
      IF(IND .EQ. MF) GO TO 100                                          026480
      MF = IND                                                           026490
      IF(IND .EQ. 1) PRINT 420                                           026500
      IF(IND .EQ. 2) PRINT 421, IFEED                                    026510
      PRINT 422                                                          026520
  420 FORMAT("0     G -- OPEN-LOOP RESPONSE"/)                           026530
  421 FORMAT("0     G/(1+",A1,"G) -- CLOSED-LOOP RESPONSE"/)             026540
  422 FORMAT(8X"W"8X,"MAGNITUDE",10X,"MAG. IN DB",6X,"ANGLE IN DEGREES") 026550
C                                                                        026560
C     COMPUTE NUMERATOR                                                  026570
C                                                                        026580
  100 CALL POLYVAL(W,A,N,PR,PI)                                          026590
      CALL POLARD(PR,PI,PMAG,PANG)                                       026600
      IF(TAUD.EQ.0.) GO TO 101                                           026610
      PANG = PANG + W*TAUD                                               026620
      PR=PMAG*COS(PANG*DEG2RAD)                                          026630
      PI=PMAG*SIN(PANG*DEG2RAD)                                          026640
  101 CONTINUE                                                           026650
      IF(PMAG .NE. 0) GO TO 140                                          026660
  102 PRINT 423,W                                                        026670
  423 FORMAT(F12.2,4X,"0.")                                              026680
      GO TO 192                                                          026690
C                                                                        026700
C     COMPUTE DENOMINATOR                                                026710
C                                                                        025720
  140 CALL POLYVAL(W,B,MD,PRD,PID)                                       026730
      IF(IND .EQ. 1) GO TO 160                                           026740
C                                                                        026750
C     CLOSED LOOP MODIFICATIONS                                          026760
C                                                                        026770
```

```
      IF(KFLAG) GO TO 158                                              026780
      CALL POLYVAL(W,E,NE,PRE,PIE)                                     026790
      CALL POLARD(PRE,PIE,EMAG,EANG)                                   026800
      CALL POLYVAL(W,F,NF,PRF,PIF)                                     026810
      CALL POLARD(PRF,PIF,FMAG,FANG)                                   026820
      IF(FMAG .NE. 0) GO TO 159                                        026830
      PMAG = 0.                                                        026840
      GO TO 102                                                        026850
  159 EMAG = (EMAG/FMAG)*PMAG                                          026850
      EANG = ((EANG-FANG)+PANG) * DEG2RAD                              026870
      PRD = PRD + EMAG*COS(EANG)                                       026880
      PID = PID + EMAG*SIN(EANG)                                       026890
      GO TO 160                                                        026900
  158 PID = PID + PI                                                   026910
      PRD = PRD + PR                                                   026920
                                                                       026930
      PRD = PRD + PR                                                   026940
  160 CALL POLARD(PRD,PID,DMAG,DANG)                                   026950
      IF(DMAG .NE. 0) GO TO 180                                        026960
      PRINT 424,W                                                      026970
  424 FORMAT(F12.2,10X,"*** POLE ***")                                 026980
      GO TO 194                                                        026990
  180 PMAG = PMAG  / DMAG                                              027000
      PANG = PANG-DANG                                                 027010
      PDB = 8.6858896 * ALOG(ABS(PMAG))                                027020
      PRINT 425,W,PMAG,PDB,PANG                                        027030
  425 FORMAT(F12.2,G18.8,G20.8,F11.4)                                  027040
  192 ICOUNT = ICOUNT + 1                                              027050
      IF(ICOUNT .GT. 500) GO TO 198                                    027060
      X(ICOUNT) = PMAG                                                 027070
```

```
      Y(ICOUNT) = PANG                                                  027080
      Z(ICOUNT)=W                                                       027090
      XDB(ICOUNT)=PDB                                                   027100
  194 W = W + DELTW                                                     027110
      IF(W-WMAX) 100,100,1                                              027120
  198 PRINT 426                                                         027130
  426 FORMAT("0TOO MANY POINTS.  EITHER PLOT WHAT YOU HAVE OR RESTART") 027140
      GO TO 1                                                           027150
  200 CALL POLPLOT                                                      027160
      GO TO 1                                                           027170
  300 LPLOT=1                                                           027180
      CALL BODP_OT                                                      027190
      GO TO 1                                                           027200
  700 PRINT 1500                                                        027210
 1500 FORMAT(//"       -------END OF PROGRAM 'FREQR'-------        "//) 027220
      IF(LPLOT.EQ.1) GO TO 701                                          027230
      RETURN                                                            027240
  701 CALL PLOTE                                                        027250
      RETURN                                                            027260
      END                                                               027270


      SUBROUTINE BODPLOT                                                027280
      COMMON/COM/X(502),Y(502),Z(502),XDB(502),ICOUNT,TTY,A(21),B(21),  027290
     1 E(21),F(21),C(40),CTEMP(40),KORD(40),D(21)                       027300
      DIMENSION TITLE(3)                                                027310
      LOGICAL TTY                                                       027320
      DATA BLANK/10H          /                                         027330
```

```
10   FORMAT(3A10)                                                          027340
15   FORMAT("0TYPE IN TITLE OF M-VS-W PLOT, 30 CHARACTERS MAX")            027350
16   FORMAT ("0TITLE IS ")                                                 027360
17   FORMAT(*0 IF YOU WANT A DB VS LOG W PLOT,ENTER A 1 *)                 027370
18   FORMAT(*0 IF YOU WANT A MAG VS W PLOT , ENTER A 0 *)                  027380
20   FORMAT(*0 IF YOU WANT BOTH TYPES OF PLOTS, ENTER A 2 *)               027390
21   FORMAT(*0 INVALID PLOT CODE, TRY AGAIN*)                              027400
22   FORMAT(*0 TYPE IN TITLE OF DB VS LOG W PLOT, 30 CHARACTERS MAX *)     027410
     IDB=0                                                                 027420
99   IF(TTY) PRINT 18                                                      027430
     IF(TTY) PRINT 17                                                      027440
     IF(TTY) PRINT 20                                                      027450
     READ *,IDB                                                            027460
     IDC=IDB                                                               027470
     IF(IDB.LT.0.OR.IDB.GT.2) GO TO 892                                    027480
     IF(ICOUNT.EQ.0) GO TO 900                                             027490
     XMAX=-1.0E10                                                          027500
     DO 30 I=1,ICOUNT                                                      027510
     IF(X(I).GT.XMAX) GO TO 25                                             027520
     GO TO 30                                                              027530
25   XMAX=X(I)                                                             027540
     XDBMAX=XDB(I)                                                         027550
     ZMX=Z(I)                                                              027560
30   CONTINUE                                                              027570
100  CONTINUE                                                              027580
     TITLE(1)=BLANK                                                        027590
     TITLE(2)=BLANK                                                        027600
     TITLE(3)=BLANK                                                        027610
     CALL PLOT(5.0,-12.0,-3)                                               027620
     CALL PLOT(0.0,2.0,-3)                                                 027630
```

```
      CALL PLOT(-.75,-.75,3)                                                    027640
      CALL PLOT(-.75,7.75,2)                                                    027650
      CALL PLOT(8.75,7.75,2)                                                    027660
      CALL PLOT(8.75,-.75,2)                                                    027670
      CALL PLOT(-.75,-.75,2)                                                    027680
      CALL PLOT(0.0,0.0,3)                                                      027690
      IF(IDB.EQ.1) GO TO 50                                                     027700
      IF(TTY) PRINT 15                                                          027710
      IF (TTY) PRINT 16                                                         027720
      READ 10,TITLE                                                             027730
      ZMIN=Z(1)                                                                 027740
      ZDA=(Z(ICOUNT)-ZMIN)/8.0                                                  027750
      Z(ICOUNT+1)=ZMIN                                                          027760
      Z(ICOUNT+2)=ZDA                                                           027770
      CALL SCALE(X,6.0,ICOUNT,1)                                                027780
      CALL SCALE(Y,6.0,ICOUNT,1)                                                027790
      XMIN=X(ICOUNT+1)                                                          027800
      XDA=X(ICOUNT+2)                                                           027810
      YMIN=Y(ICOUNT+1)                                                          027820
      YDA=Y(ICOUNT+2)                                                           027830
      CALL AXIS(0.0,0.0,11HFREQUENCY W,-11,8.0,0.0,ZMIN,ZDA)                    027840
      CALL AXIS(0.0,0.0,4HM(W),4,6.,90.,XMIN,XDA)                               027850
      CALL AXIS(8.0,0.0,8HANGLE(W),-8,6.,90.,YMIN,YDA)                          027860
      CALL LINE(Z,X,ICOUNT,1,0,0)                                               027870
      CALL LINE(Z,Y,ICOUNT,1,10,5)                                              027880
      GO TO 70                                                                  027890
50    CONTINUE                                                                  027900
      IDB=3                                                                     027910
      IF(TTY) PRINT 22                                                          027920
      IF(TTY) PRINT 16                                                          027930
```

```
      READ 10,TITLE                                                    027940
      XMAX=XDBMAX                                                      027950
      CALL SCALE(XDB,6.,ICOUNT,1)                                      027960
      IF(IDC.EQ.1) CALL SCALE(Y,6.0,ICOUNT,1)                          027970
      CALL LGSCAL(Z,8.,ICOUNT)                                         027980
      ZMIN=Z(ICOUNT+1)                                                 027990
      ZDA=Z(ICOUNT+2)                                                  028000
      XMIN=XDB(ICOUNT+1)                                               028010
      XDA=XDB(ICOUNT+2)                                                028020
      YMIN=Y(ICOUNT+1)                                                 028030
      YDA=Y(ICOUNT+2)                                                  028040
      CALL LGAXIS(0.0,0.0,11HFREQUENCY W,-11,8.,0.,ZMIN,ZDA)           028050
      CALL AXIS(0.,0.,7HM(W) DB,7,6.,90.,XMIN,XDA)                     028060
      CALL AXIS(8.,0.,8HANGLE(W),-8,6.,90.,YMIN,YDA)                   028070
      CALL LGLINE(Z,XDB,ICOUNT,0,0,-5)                                 028080
      CALL LGLINE(Z,Y,ICOUNT,10,5,-5)                                  028090
70    CONTINUE                                                         028100
      CALL SYMBOL(1.4,7.0,0.21,TITLE,0.0,30)                           028110
      CALL SYMBOL(6.0,6.24,0.14,11H-=MAGNITUDE,0.0,11)                 028120
      CALL SYMBOL(6.0,6.57,0.14,5,0.0,-1)                              028130
      CALL SYMBOL(6.14,6.5,0.14,6H=ANGLE,0.0,6)                        028140
      CALL SYMBOL(1.0,6.5,0.14,3HM =,0.0,3)                            028150
      CALL NUMBER(999.,999.,0.14,XMAX,0.0,4)                           028160
      IF(IDB.EQ.3) CALL SYMBOL(999.,999.,0.14,3H DB,0.0,3)             028170
      CALL SYMBOL(1.14,6.465,0.07,1HM,0.0,1)                           028180
      CALL SYMBOL(1.0,6.24,0.14,11H,0.0,-1)                            028190
      CALL SYMBOL(1.14,6.205,0.07,1HM,0.0,1)                           028200
      CALL SYMBOL(1.28,6.24,0.14,1H=,0.0,1)                            028210
      CALL NUMBER(999.,999.,0.14,ZMX,0.0,4)                            028220
      CALL PLOT(9.0,0.0,-3)                                            028230
```

```
      IF(IDB.EQ.2) GO TO 800                                            028240
      IF(IDB.EQ.3) GO TO 890                                            028250
      GO TO 890                                                         028260
  800 IDB=1                                                             028270
      GO TO 100                                                         028280
  892 IF(TTY) GO TO 893                                                 028290
      PRINT 21                                                          028300
      GO TO 890                                                         028310
  893 PRINT 21                                                          028320
      GO TO 99                                                          028330
  890 ICOUNT =0                                                         028340
  900 CONTINUE                                                          028350
      RETURN                                                            028360
      END                                                               028370
```



```
      SUBROUTINE POLPLOT                                                028380
      COMMON/COM/X(502),Y(502),Z(502),XD3(502),ICOUNT,TTY,A(21),B(21),  028390
     1 E(21),F(21),C(40),CTEMP(40),KORD(40),D(21)                       028400
      EQUIVALENCE (X(1),IX(1)),(Y(1),IY(1))                             028410
      COMMON /CONVERT/ RAD2DEG,DEG2RAD                                  028420
      DIMENSION IX(500),IY(500),LINE(121)                               028430
      INTEGER TOPFORM(3),BOTFORM(3)                                     028440
      LOGICAL TTY                                                       028450
      INTEGER DOT,ZERO,BLANK                                            028460
      DATA DOT/1H./, BLANK/1H /, ZERO/1H0/, STAR/1H*/                   028470
      IF(ICOUNT .EQ. 0) GO TO 20                                        028480
C                                                                       028490
```

```
C     CONVERT TO RECTANGULAR COORDINATES                          028500
C     AT THE SAME TIME FIND MAX AND MIN VALUES                    028510
C                                                                 028520
      XMAX = XMIN = YMAX = YMIN = 0                               028530
      DO 1 I=1,ICOUNT                                             028540
      YRAD = Y(I)*DEG2RAD                                         028550
      XTEMP = X(I)*COS(YRAD)                                      028560
      YTEMP = X(I)*SIN(YRAD)                                      028570
      XMAX = AMAX1(XTEMP,XMAX)                                    028580
      XMIN = AMIN1(XTEMP,XMIN)                                    028590
      YMAX = AMAX1(YTEMP,YMAX)                                    028600
      YMIN = AMIN1(YTEMP,YMIN)                                    028610
      X(I) = XTEMP                                                028620
    1 Y(I) = YTEMP                                                028630
C                                                                 028640
C     COMPUTE SCALE FACTORS                                       028650
C                                                                 028660
      XRAN = XMAX - XMIN                                          028670
      YRAN = YMAX - YMIN                                          028680
      RANGE = AMAX1(XRAN,YRAN)                                    028690
      XBIAS = 31.0                                                028700
      YBIAS = 19.0                                                028710
      IXMAX = 61                                                  028720
      IYMAX = 37                                                  028730
      IF(TTY) GO TO 3                                             028740
      XBIAS = 61.0                                                028750
      YBIAS = 37.0                                                028760
      IXMAX = 121                                                 028770
      IYMAX = 73                                                  028780
    3 XFACT = IXMAX/RANGE                                         028790
```

```
      YFACT = IYMAX/RANGE                                              028800
      XBIAS = XBIAS -(XMIN + .5*XRAN)*XFACT                            028810
      YBIAS = YBIAS - (YMIN + .5*YRAN)*YFACT                           028820
      IY1 = IYMAX + 1                                                  028830
      DO 4 I=1,ICOUNT                                                  028840
      IXX = X(I)*XFACT + XBIAS                                         028850
      IF(IXX .LT. 1) IXX = 1                                           028860
      IF(IXX .GT. IXMAX) IXX = IXMAX                                   028870
      IX(I) = IXX                                                      028880
      IYY = Y(I)*YFACT + YBIAS                                         028890
      IF(IYY .LT. 1) IYY = 1                                           028900
      IF(IYY .GT. IYMAX) IYY = IYMAX                                   028910
C        Y MUST BE INVERTED SINCE WE PLOT FROM THE TOP DOWN             028920
    4 IY(I) = IY1 - IYY                                                028930
C                                                                      028940
C     SORT X,Y PAIRS ON Y                                              028950
C     USES A SHELL SORT                                                028960
C                                                                      028970
      IF(ICOUNT .LE. 1) GO TO 107                                      028980
      ID = 1                                                           028990
  110 ID = SHIFT(ID,1)                                                 029000
      IF(ID .LE. ICOUNT) GO TO 110                                     029010
      ID = SHIFT(ID,-1)-1                                              029020
  101 DO 106 I=1,ID                                                    029030
      J1 = I                                                           029040
      J2 = I+ID                                                        029050
      GO TO 105                                                        029060
  102 IF(IY(J1) .LE. IY(J2)) GO TO 104                                 029070
      J3 = J1                                                          029080
      J4 = J2                                                          029090
```

```
  103  ITEMP = IY(J3)                                                  029100
       IY(J3) = IY(J4)                                                 029110
       IY(J4) = ITEMP                                                  029120
       ITEMP = IX(J3)                                                  029130
       IX(J3) = IX(J4)                                                 029140
       IX(J4) = ITEMP                                                  029150
       J4 = J3                                                         029160
       J3 = J3-ID                                                      029170
       IF(J3 .LT. 1) GO TO 104                                         029180
       IF(IY(J3) .GT. IY(J4)) GO TO 103                                029190
  104  J1 = J2                                                         029200
       J2 = J2 + ID                                                    029210
  105  IF(J2 .LE. ICOUNT) GO TO 102                                    029220
  106  CONTINUE                                                        029230
       ID = ID/2                                                       029240
       IF(ID .GE. 1) GO TO 101                                         029250
C                                                                      029260
C      DO ACTUAL PLOT                                                  029270
C                                                                      029280
  107  IXZ = XBIAS                                                     029290
       IF(IXZ .LT. 1) IXZ = 1                                          029300
       IF(IXZ .GT. IXMAX) IXZ = IXMAX                                  029310
       IYZ = YBIAS                                                     029320
       IF(IYZ .LT. 1) IYZ = 1                                          029330
       IF(IYZ .GT. IYMAX) IYZ = IYMAX                                  029340
       IYZ = IY1 - IYZ                                                 029350
       I1MAX = IYZ-1                                                   029360
       I2MIN = IYZ+1                                                   029370
       IXZZ = IXZ - 4                                                  029380
       IF(IXZZ .LT. 0) IXZZ = 0                                        029390
```

```
      ENCODE(23,5,TOPFORM)IXZZ                                          029400
      PRINT TOPFORM                                                     029410
5     FORMAT(5H("1",,I5,134X,"-270 DEG"))                               029420
      PRINT 6,(STAR,I=1,IXMAX),STAR                                     029430
6     FORMAT(" *",122A1)                                                029440
      IC = 1                                                            029450
      IF(I1MAX .EQ. 0) GO TO 10                                         029460
      DO 9 I=1,I1MAX                                                    029470
      DO 7 J=1,IXMAX                                                    029480
7     LINE(J) = BLANK                                                   029490
      LINE(IXZ) = DOT                                                   029500
8     IF(IC .GT. ICOUNT) GO TO 9                                        029510
      IF(IY(IC) .GT. I) GO TO 9                                         029520
      LINE(IX(IC)) = ZERO                                               029530
      IC = IC + 1                                                       029540
      GO TO 8                                                           029550
9     PRINT 6,(LINE(J),J=1,IXMAX),STAR                                  029560
10    DO 11 J=1,IXMAX                                                   029570
11    LINE(J) = DOT                                                     029580
12    IF(IC .GT. ICOUNT) GO TO 13                                       029590
      IF(IY(IC) .GT. IYZ) GO TO 13                                      029600
      LINE(IX(IC)) = ZERO                                               029610
      IC = IC + 1                                                       029620
      GO TO 12                                                          029630
13    PRINT 6,(LINE(J),J=1,IXMAX),STAR                                  029640
      IF(I2MIN .GT. IYMAX) GO TO 17                                     029650
      DO 16 I=I2MIN,IYMAX                                               029660
      DO 14 J=1,IXMAX                                                   029670
14    LINE(J) = BLANK                                                   029680
      LINE(IXZ) = DOT                                                   029690
```

```
15    IF(IC .GT. ICOUNT) GO TO 16                                    029700
      IF(IY(IC) .GT. I) GO TO 15                                     029710
      LINE(IX(IC)) = ZERO                                            029720
      IC = IC + 1                                                    029730
      GO TO 15                                                       029740
16    PRINT 6,(LINE(J),J=1,IXMAX),STAR                               029750
17    PRINT 6,(STAR,J=1,IXMAX),STAR                                  029760
      ENCODE(26,18,BOTFORM)IXZZ                                      029770
      PRINT BOTFORM                                                  029780
18    FORMAT(5H(" ",,I5,15HX,"-30 DEG"/"1"))                         029790
      ICOUNT = 0                                                     029800
      RETURN                                                         029810
20    PRINT 21                                                       029820
21    FORMAT("-NO POINTS COMPUTED - NO PLOT AVAILABLE")              029830
      RETURN                                                         029840
      END                                                            029850


      SUBROUTINE POLYVAL(W,COEF,ORDER,PR,PI)                         029860
      INTEGER ORDER                                                  029870
      DIMENSION COEF(1)                                              029880
      Z = -W*W                                                       029890
      IF((ORDER .AND. 1) .EQ. 0) GO TO 6                             029900
C                                                                    029910
C                  ORDER IS ODD                                      029920
C                                                                    029930
      PR = COEF(2)                                                   029940
      PI = COEF(1)                                                   029950
```

```
      IF(ORDER .EQ. 1) GO TO 4                                          029960
      DO 2  I=3,ORDER,2                                                 029970
      PR = PR*Z + COEF(I+1)                                             029980
    2 PI = PI*Z + COEF(I)                                               029990
    4 PI = PI*W                                                         030000
      RETURN                                                            030010
C                                                                       030020
C                     ORDER IS EVEN                                     030030
C                                                                       030040
    6 IF (ORDER .EQ. 0) GO TO 12                                        030050
      PR = COEF(1)*Z + COEF(3)                                          030060
      PI = COEF(2)                                                      030070
      IF (ORDER .EQ. 2) GO TO 10                                        030080
      DO 8  I=4,ORDER,2                                                 030090
      PR = PR*Z + COEF(I+1)                                             030100
    8 PI = PI*Z + COEF(I)                                               030110
   10 PI = PI*W                                                         030120
      RETURN                                                            030130
C                                                                       030140
C                     ORDER IS 0                                        030150
C                                                                       030160
   12 PI = 0.                                                           030170
      PR = COEF(1)                                                      030180
      RETURN                                                            030190
      END                                                               030200


      SUBROUTINE POLARD(REAL,IMAG,MAG,ANG)                              030210
```

```
C      .                                                              030220
C      COMPUTES ANGLE  IN RANGE 0-360                                 030230
C                                                                     030240
       COMMON /CONVERT/ RAD2DEG,DEG2RAD                               030250
       REAL MAG,IMAG                                                  030260
       MAG = SQRT(REAL**2 + IMAG**2)                                  030270
       IF(REAL) 5,1,5                                                 030280
 1     IF(IMAG)2,3,4                                                  030290
 2     ANG = 270.                                                     030300
       RETURN                                                         030310
 3     ANG = 0.                                                       030320
       RETURN                                                         030330
 4     ANG = 90.                                                      030340
       RETURN                                                         030350
 5     ANG = ATAN(IMAG/REAL)*RAD2DEG                                  030360
       IF(ANG .LT. 0) ANG = ANG + 360.                                030370
       RETURN                                                         030380
 5     ANG = ATAN(IMAG/REAL)*RAD2DEG + 180.                           030390
       RETURN                                                         030400
       END                                                            030410


       SUBROUTINE POLYRD(A,N,B,MD,OGN)                                030420
       DIMENSION A(21),B(21)                                          030430
       COMMON/COM/X(502),Y(502),Z(502),XDB(502),ICOUNT,TTY,DUM(42),   030440
      1 E(21),F(21),C(40),CTEMP(40), KORD(40),D(21)                   030450
       LOGICAL TTY,MM                                                 030460
       IF(TTY) PRINT 35                                               030470
```

```
35    FORMAT("0TYPE GAIN CONSTANT K -- ")                              030480
      READ *,D(1)                                                      030490
      PRINT 36,D(1)                                                    030500
      OGN=D(1)                                                         030510
      FBOG=D(1)                                                        030520
36    FORMAT("0GAIN CONSTANT =",G15.7)                                 030530
      MULT = 1                                                         030540
      IF(TTY) PRINT 1                                                  030550
1     FORMAT("0TYPE ORDER OF NUMERATOR -- ")                           030560
 4    READ *,FMD                                                       030570
      MD = FMD                                                         030580
      IF(MD .GE. 0 .AND. MD .LE. 20) GO TO 5                           030590
2     PRINT 3                                                          030600
3     FORMAT("0ERROR IN INPUT DATA,  RUN TERMINATED.")                 030610
      STOP                                                             030620
5     MM = .TRUE.                                                      030630
      ND = 1                                                           030640
6     IF(TTY) PRINT 7                                                  030650
7     FORMAT("0TYPE NUMBER OF FACTORS OF A GIVEN ORDER -- ")           030660
      READ *,FNUM                                                      030670
      NUM = FNUM                                                       030680
      IF(TTY) PRINT 9                                                  030690
9     FORMAT("0TYPE ORDER OF THOSE FACTORS -- ")                       030700
      READ *,FMORD                                                     030710
      MORD = FMORD                                                     030720
      MORD1 = MORD + 1                                                 030730
      K = NUM*MORD1                                                    030740
      K1 = K + 1                                                       030750
      IF(TTY) PRINT 10 ,K                                              030760
10    FORMAT(" TYPE",I3," COEFFICIENT(S)"/)                            030770
```

```
      DO 12 I=1,K                                                       030780
      READ *,CTEMP(I)                                                   030790
      C(K1-I) = CTEMP(I)                                                030800
12    CONTINUE                                                          030810
      IF(.NOT.(TTY.OR.MM))GO TO 16                                      030820
      IF(MULT .EQ. 1) PRINT 13                                          030830
13    FORMAT("0COEFFICIENT(S) OF NUMERATOR")                            030840
      IF (MULT .NE. 1) PRINT 14                                         030850
14    FORMAT("0COEFFICIENT(S) OF DENOMINATOR")                          030860
      MM = .FALSE.                                                      030870
16    DO 18 I=1,MORD1                                                   030880
18    KORD(I) = I-1                                                     030890
      DO 22 J=1,K,MORD1                                                 030900
      JEND = J + MORD                                                   030910
      PRINT 19 ,(CTEMP(I),KORD(JEND+1-I),I=J,JEND)                      030920
19    FORMAT(" ",3(1PE13.6," S(",I2,") ")/(21X,E13.6," S(",I2,") ",     030930
     +   E13.6," S(",I2,")"))                                           030940
      CALL POLYMUL(C(J),MORD1,D,ND,B,NB)                                030950
      DO 20 L=1,NB                                                      030960
20    D(L) = B(L)                                                       030970
22    ND = NB                                                           030980
      IF(MD-ND+1) 2,24,6                                                030990
24    IF(MULT .NE. 1) GO TO 30                                          031000
      DO 28 I=1,ND                                                      031010
28    A(I) = D(ND+1-I)                                                  031020
      N = MD                                                            031030
      MULT = 2                                                          031040
      IF(TTY) PRINT 29                                                  031050
29    FORMAT("0TYPE ORDER OF DENOMINATOR -- ")                          031060
      D(1) = 1.                                                         031070
```

```
      GO TO  4                                                          031080
30    DO 32 I=1,ND                                                      031090
32    B(I) = D(ND+1-I)                                                  031100
      RETURN                                                            031110
      END                                                               031120


      SUBROUTINE POLYMUL(A,NA,B,NB,C,NC)                                031130
      DIMENSION A(1),B(1),C(1)                                          031140
      NC = NA+NB-1                                                      031150
      DO 1 I=1,NA                                                       031160
1     C(NB-1+I) = A(I)*B(NB)                                            031170
      IF(NB .EQ. 1) RETURN                                              031180
      NB1 = NB-1                                                        031190
      DO 2 I=1,NB1                                                      031200
2     C(I) = A(1)*B(I)                                                  031210
      IF(NA .EQ. 1) RETURN                                              031220
      DO 3 I=2,NA                                                       031230
      DO 3 J=1,NB1                                                      031240
3     C(I+J-1) = C(I+J-1) + A(I)*B(J)                                   031250
      RETURN                                                            031260
      END                                                               031270


      OVERLAY(COPCSD,3,0)                                               031280
      PROGRAM PARTL                                                     031290
```

```
      COMMON /POLARC/ AC,BD,FACT,FACTR                                  031300
      COMMON Z(12),XR,NXX,Y(24),W(24),CR(24),EC(24),OM(24),FE(24),L     031310
      INTEGER DOT,BLANK,CROSS,COLON,TLINE                               031320
      LOGICAL OVPAGE,TELL                                               031330
      DIMENSION A(24),B(24),C(24),D(24),E(24),F(24),P(12),Q(12)         031340
      DIMENSION TLINE(102),FTOR(12),X(503),Y1(503),TITLE(5)             031350
      DATA FACTR/1./,FACT/0./,PI/3.1415927/,DEGRE/57.29578/             031360
      DATA PI2/6.283185307/                                             031370
      DATA DOT/1H./,BLANK/1H /,CROSS/1HX/,COLON/1H:/,IZERO/1H0/         031380
      DATA FTOR/1.,1.,2.,6.,24.,120.,720.,5040.,40320.,                 031390
     +   362880.,3628800.,39916800./                                    031400
      DATA TELL/.TRUE./                                                 031410
C   INITIALIZE ARRAYS                                                   031420
C   OVERRUN ON Z(I) ALSO SETS XR AND NXX                                031430
 1    DO 2   I=1,24                                                    031440
 2    Y(I)=W(I)=CR(I)=OM(I)=FE(I)=EC(I)=Z(I)=0                         031450
      H=0                                                               031460
      FIN=0.                                                            031470
C                                                                       031480
C     INPUT SECTION                                                     031490
C                                                                       031500
C     K = NUMBER OF ZEROS                                               031510
C     L = NUMBER OF NON-REPEATED POLES IN DENOMINATOR                   031520
C     M = POWER OF REPEATED POLE IN DENOMINATOR                         031530
C     CONST = CONSTANT IN NUMERATOR                                     031540
C                                                                       031550
      WRITE(7,300)                                                      031560
 6    WRITE(7,223)                                                      031570
      READ *,FK,FL,FM,CONST                                             031580
      K = FK   $ L = FL   $ M = FM                                      031590
```

```
C                                                                       031600
C     TEST CONSTRAINTS ON K, L, AND M                                   031610
C                                                                       031620
      IF(K .LT. 0  .OR.  K .GT. 12) GO TO 8                             031630
      IF(L .LT. 0  .OR.  L .GT. 24) GO TO 8                             031640
      IF(M .LT. 0  .OR. M .EQ. 1  .OR.  M .GT. 12) GO TO 8              031650
      IF(K .GE. L + M) GO TO 8                                          031660
      IF(K .GT. L + 1) GO TO 8                                          031670
      GO TO 10                                                          031680
C                                                                       031690
C     ILLEGAL INPUT                                                     031700
C                                                                       031710
 8    WRITE(7,280)                                                      031720
      GO TO 6                                                           031730
C                                                                       031740
C     ACCEPT NUMERATOR VALUES                                           031750
C                                                                       031760
 10   IF(K .EQ. 0) GO TO 18                                             031770
 12   JY=0                                                              031780
 14   JY=JY+1                                                           031790
      WRITE(7,224)                                                      031800
      READ *,A(JY),B(JY)                                                031810
      IF(B(JY).EQ.0) GO TO 16                                           031820
      JY=JY+1                                                           031830
      A(JY)=A(JY-1)                                                     031840
      B(JY)=B(JY-1)                                                     031850
      WRITE(7,255)                                                      031860
 16   IF(JY.LT.K)GO TO 14                                               031870
C                                                                       031880
C     ACCEPT NON REPEATED VALUES IN DENOMINATOR                         031890
```

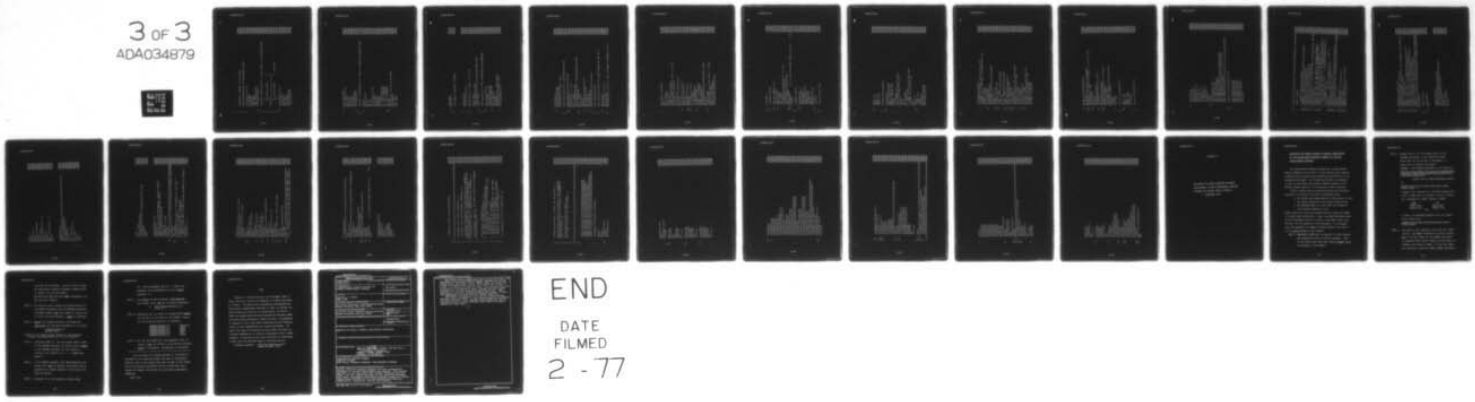
```
C                                                                       031900
 18   IF(L .EQ. 0) GO TO 28                                             031910
      JX=0                                                              031920
 20   JX=JX+1                                                           031930
      WRITE(7,225)                                                      031940
      READ *,C(JX),D(JX)                                                031950
      IF(D(JX).EQ.0) GO TO 22                                           031960
      JX=JX+1                                                           031970
      C(JX)=C(JX-1)                                                     031980
      D(JX)=-D(JX-1)                                                    031990
      WRITE(7,255)                                                      032000
 22   IF(JX.LT.L)GO TO 20                                               032010
C                                                                       032020
C     THIS ALGORITHM BASED ON A PAPER BY DOV HAZONY AND JACK RILEY      032030
C     PRESENTED ON 21 AUG, 1959 AT THE AUTOMATIC CONTROL SESSION OF THE 032040
C     WESTERN ELECTRONIC SHOW AND CONVENTION                            032050
C                                                                       032060
C     THIS SECTION SETS (E(N),F(N)) AS POLAR COORDINATE FORM OF         032070
C                                                                       032080
C                           K     DO 30          !                      032090
C          (S+C(N)+JD(N))   TT (S+A(I)+JB(I))    !                      032100
C                           I=1                   !                      032110
C     R(N1)=----------------------------------------!                     032120
C                           L                     !                      032130
C          (S+G+JH)**M      TT (S+C(J)+JD(J))     ! S=-(C(N)+JD(N))      032140
C            DO 38          J=1     DO 34         !                      032150
C                                                                       032160
C         =E(N)+F(N)                                                    032170
C                                                                       032180
 28   IF(L.EQ.0)GO TO 36                                                032190
```

```
      DO 34 N=1,L                                                    032200
      E(N)=1.                                                        032210
      F(N)=0.                                                        032220
      IF(K .EQ. 0) GO TO 32                                          032230
      DO 30 I=1,K                                                    032240
      AC=A(I)-C(N)                                                   032250
      BD=B(I)-D(N)                                                   032260
      CALL POLAR                                                     032270
      F(N)=F(N)+FACT                                                 032280
30    E(N)=E(N)*FACTR                                                032290
32    E(N)=E(N)*CONST                                                032300
      DO 34 J=1,L                                                    032310
      IF( J-N  .EQ. 0) GO TO 34                                      032320
      AC=C(J)-C(N)                                                   032330
      BD=D(J)-D(N)                                                   032340
      CALL POLAR                                                     032350
      E(N)=E(N)/FACTR                                                032360
      F(N)=F(N)-FACT                                                 032370
34    CONTINUE                                                       032380
C                                                                    032390
C     PARTIAL FRACTION EXPANSION IS COMPLETE IF M = 0                032400
C                                                                    032410
36    IF(M.EQ.0)GO TO 56                                             032420
C                                                                    032430
C     ACCEPT REPEATED VALUES                                         032440
C                                                                    032450
      WRITE(7,226)                                                   032460
      READ *,G                                                       032470
      P(M) = 1.                                                      032480
      Q(M)=0.                                                        032490
```

AD-A034 879 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPC--ETC(U)
DEC 76 F L O'BRIEN

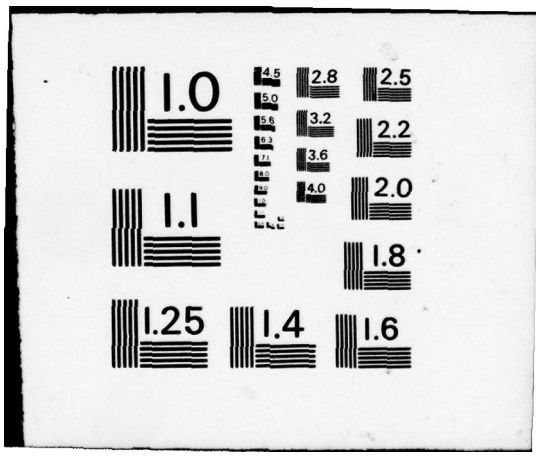
UNCLASSIFIED GE/EE/76D-38 NL

3 OF 3
ADA034879



END

DATE
FILMED
2 - 77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

```
C                                                                       032500
C     PROCEDURE TO HANDLE REPEATED ROOT                                 032510
C                                                                       032520
C     MODIFY COEFFICIENTS OF NON-REPEATED ROOTS                         032530
C                                                                       032540
      IF(L.EQ.0)GO TO 40                                                032550
      DO 38 N=1,L                                                       032560
      AC=G-C(N)                                                         032570
      BD=H-D(N)                                                         032580
      CALL POLAR                                                        032590
      E(N)=E(N)/FACTR**M                                                032600
   38 F(N)=F(N)-FACT*M                                                  032610
C                                                                       032620
C     GENERATE COEFFICIENT OF THE M*TH POWER OF THE REPEATED ROOT       032630
C                                                                       032640
C          K                          !                                032650
C          TT (S+A(I)+JB(I))          !                                032660
C          I=1                        !                                032670
C     C(M)=---------------------------!                                032680
C          L                          !                                032690
C          TT (S+C(J)+JD(J))          !  S=-(G+JH)                     032700
C          J=1                        !                                032710
C                                                                       032720
   40 IF(K.EQ.0)GO TO 44                                                032730
      DO 42 I=1,K                                                       032740
      AC=A(I)-G                                                         032750
      BD=B(I)-H                                                         032760
      CALL POLAR                                                        032770
      P(M)=P(M)*FACTR                                                   032780
   42 Q(M)=Q(M)+FACT                                                    032790
```

```
 44   P(M)=P(M)*CONST                                                    032800
      IF(L.EQ.0)GO TO 48                                                 032810
      DO 46 J=1,L                                                        032820
      AC=C(J)-G                                                          032830
      BD=D(J)-H                                                          032840
      CALL POLAR                                                         032850
      P(M)=P(M)/FACTR                                                    032860
 46   Q(M)=Q(M)-FACT                                                     032870
C                                                                        032880
C     GENERATE COEFFICIENTS FOR 1 THRU M-1 POWERS OF REPEATED ROOT       032890
C                                                                        032900
 48   MI=M-1                                                             032910
      DO 54 N=1,MI                                                       032920
      NA=N-1                                                             032930
      AA = 0                                                             032940
      AB = 0                                                             032950
      IF(L.EQ.0)GO TO 52                                                 032960
      DO 50 J=1,L                                                        032970
      AC=G-C(J)                                                          032980
      BD=H-D(J)                                                          032990
      CALL POLAR                                                         033000
      R = E(J)*FACTR**NA                                                 033010
      S = F(J) + FACT*NA                                                 033020
      AA = AA + R*COS(S)                                                 033030
 50   AB = AB + R*SIN(S)                                                 033040
 52   DELTA=0                                                            033050
      IF( M+L-<-N  .EQ. 0) DELTA=CONST                                   033060
      AC=DELTA-AA                                                        033070
      BD = -AB                                                           033080
      CALL POLAR                                                         033090
```

```
      P(N)=FACTR                                                        033100
   54 Q(N)=FACT                                                         033110
C                                                                       033120
C     END PARTIAL FRACTION EXPANSION                                    033130


C                                                                       033140
   56 WRITE(6,200)                                                      033150
      IF(L.EQ.0)GO TO 60                                                033160
C                                                                       033170
C     REDUCE ANGLES TO RANGE -PI TO +PI                                 033180
C                                                                       033190
      DO 58 J=1,L                                                       033200
      TEMP = AMOD(F(J),PI2)                                             033210
      IF(ABS(TEMP).GT. PI) TEMP=TEMP-SIGN(PI2,TEMP)                     033220
   58 F(J) = TEMP                                                       033230
C                                                                       033240
C     PRINT ROOTS WITH THEIR COEFFICIENTS (RECTANGULAR)                 033250
C                                                                       033260
   60 IF(M.EQ.0)GO TO 66                                                033270
      DO 62 N=1,M                                                       033280
      A(N)=P(N)*COS(Q(N))                                               033290
   62 B(N)=P(N)*SIN(Q(N))                                               033300
      DO 64 NAA=1,M                                                     033310
      N=M+1-NAA                                                         033320
      IF(ABS(B(N)).LE..0001) B(N)=0                                     033330
   64 WRITE(6,203)G,H,N,A(N),B(N)                                       033340
      IF(L.EQ.0)GO TO 70                                                033350
```

```
 66   DO 68 N=1,L                                                    033350
      A(N)=E(N)*COS(F(N))                                            033370
      B(N)=E(N)*SIN(F(N))                                            033380
      IF(ABS(B(N)).LE..0001) B(N)=0                                  033390
C                                                                    033400
C     PRINT ROOTS WITH THEIR COEFFICIENTS (POLAR)                    033410
C                                                                    033420
 68   WRITE(6,204)C(N),D(N),A(N),B(N)                                033430
 70   WRITE(6,201)                                                   033440
      IF (M .EQ. 0) GO TO 76                                         033450
      DO 72 J=1,M                                                    033460
      TEMP = AMOD(Q(J),PI2)                                          033470
      IF(ABS(TEMP) .GT. PI) TEMP=TEMP-SIGN(PI2,TEMP)                 033480
 72   A(J) = TEMP*DEGRE                                              033490
      DO 74 JA=1,M                                                   033500
      J=M+1-JA                                                       033510
 74   WRITE(6,202)G,H,J,P(J),A(J)                                    033520
      IF(L.EQ.0)GO TO 80                                             033530
 76   DO 78 N=1,L                                                    033540
      B(N)=F(N)*DEGRE                                                033550
 78   WRITE(6,205)C(N),D(N),E(N),B(N)                                033560
C                                                                    033570
C     TIME FUNCTION OUTPUT                                           033580
C                                                                    033590
C     THIS SECTION ALSO SETS UP BLANK COMMON FOR USE BY FT           033600
C                                                                    033610
 80   WRITE(7,228)                                                   033620
      IF (M .EQ. 0) GO TO 88                                         033630
C     REAL REP ROOT                                                  033640
C     MULTIPLICATION BY ONE IS USED TO CHANGE A POSSIBLE -0 TO 0     033650
```

```
C       FOR PRINTING PURPOSES                                          033660
        NXX=1                                                          033670
        XR = 1.*(-G)                                                   033680
        DO 86  JX=1,M                                                  033690
        JY=M+1-JX                                                      033700
        Z(JY) = P(JY)/FTOR(JY)                                         033710
        IF(ABS(A(JY)) .GE. 170.) Z(JY) = -Z(JY)                        033720
        IF(JY .LE. 2) GO TO 82                                         033730
        JY = JY - 1                                                    033740
        WRITE(7,229) Z(JY+1),JY,XR                                     033750
        GO TO 86                                                       033760
 82     IF(JY .EQ. 1) GO TO 84                                         033770
        WRITE(7,237)Z(2),XR                                            033780
        GO TO 85                                                       033790
 84     WRITE(7,230)Z(1),XR                                            033800
 85     CONTINUE                                                       033810
 88     IF (L.EQ.0) GO TO 96                                           033820
        DO 94  NR=1,L                                                  033830
        IF (D(NR))90,92,94                                             033840
 90     CR(NR)=2.*E(NR)                                                033850
        EC(NR) = 1.*(-C(NR))                                           033860
        OM(NR) = 1.*(-D(NR))                                           033870
        FE(NR)=B(NR)+90.                                               033880
        WRITE(7,231)CR(NR),EC(NR),OM(NR),FE(NR)                        033890
        GO TO 94                                                       033900
 92     Y(NR) = E(NR)                                                  033910
        IF(ABS(B(NR)) .GE. 170.) Y(NR) = -E(NR)                        033920
        W(NR) = 1.*(-C(NR))                                            033930
        WRITE(7,230)Y(NR),W(NR)                                        033940
        IF(C(NR).EQ.0.)FIN=FIN+Y(NR)                                   033950
```

```
 94   CONTINUE                                                          033960
C                                                                      033970
C     OPTIONS PORTIONS                                                 033980
C                                                                      033990
 95   IF(TELL) WRITE(7,227)                                             034000
      TELL = .FALSE.                                                   034010
 98   WRITE(7,250)                                                      034020
      READ *,FNIX                                                      034030
      NIX = FNIX                                                       034040
      IF((NIX.LT.0.).OR.NIX.GT.5) GO TO 98                             034050
      IF(NIX.EQ.0.) GO TO 100                                          034060
      GO TO (102,106,146,148,1)NIX                                     034070
 100  WRITE(7,3000)                                                     034080
 3000 FORMAT(//"       --------END OF PROGRAM 'PARTL'--------      "//) 034090
      RETURN                                                           034100
C                                                                      034110
C     TABULAR LISTING                                                  034120
C                                                                      034130
 102  WRITE(7,250)                                                      034140
      READ *,T,DUR,DT                                                  034150
      N=DUR/DT+1.                                                      034160
      IF(N.GT.200)N=200                                                034170
      WRITE(6,248)                                                     034180
      DO 104 I=1,N                                                     034190
      FU=FT(T)                                                         034200
      WRITE(6,235)T,FU                                                 034210
 104  T=T+DT                                                           034220
      GO TO 95                                                         034230
C                                                                      034240
C     SET UP FOR TTY PLOT                                              034250
```

```
C                                                                       034260
 106  NTOP = 50                                                         034270
      NFILE = 7                                                         034280
      NLINE = 61                                                        034290
      BIASL = 50.                                                       034300
C                                                                       034310
C     PRINTED PLOT GENERATOR                                            034320
C                                                                       034330
 108  WRITE(7,234)                                                      034340
      READ *,T,DUR,DT,BIAS                                              034350
      BIASP = BIAS + 1.5                                                034360
C     RECORD ORIGINAL TIME                                              034370
      TORG=T                                                            034380
      N = (DUR/DT) + 1.                                                 034390
C     LIMIT NUMBER OF POINTS PLOTTED                                    034400
      IF (N .GT. NTOP) N = NTOP                                         034410
      FMAX=0.                                                           034420
      TP=0.                                                             034430
      DO 110 I=1,N                                                      034440
      FU = ABS(FT(T))                                                   034450
C      TP WILL BE TIME TO THE PEAK VALUE                                034460
       IF(FMAX .GE. FU) GO TO 110.                                      034470
      TP = T                                                            034480
      FMAX = FU                                                         034490
 110  T=T+DT                                                            034500
C       RESET TIME                                                      034510
      T=TORG                                                            034520
      SGNE=FT(TP)                                                       034530
      WRITE(NFILE,245)FIN                                               034540
      WRITE(NFILE,246)TP                                                034550
```

```
      WRITE(NFILE,249)SGNE                                              034560
C     IF ORIGIN IS ON PAGE MARK IT WITH A ZERO                          034570
      IF(BIASP .LT. 1 .OR. BIAS .GT. BIASL) GO TO 114                   034580
      ONPAGE = .TRUE.                                                   034590
      DO 112 J=1,NLINE                                                  034600
  112 TLINE(J)=BLANK                                                    034610
      TLINE(BIASP)=IZERO                                                034620
      WRITE(NFILE,240)(TLINE(J),J=1,NLINE)                              034630
      GO TO 122                                                         034640
  114 ONPAGE = .FALSE.                                                  034650
      IF(SGNE) 116,116,118                                              034660
  116 FLOW = -FMAX                                                      034670
      FHIGH = (BIASL-BIAS)*FMAX/BIAS                                    034680
      GO TO 120                                                         034690
  118 FLOW = -BIAS*FMAX/(BIASL-BIAS)                                    034700
      FHIGH = FMAX                                                      034710
  120 WRITE(NFILE,295)FLOW,FHIGH                                        034720
C     MAKE Y AXIS WITH DOTS                                             034730
  122 DO 124 J=1,NLINE                                                  034740
  124 TLINE(J)=DOT                                                      034750
      WRITE(NFILE,240)(TLINE(J),J=1,NLINE)                              034760
C     COMPUTE THE MAX AT THE PEAK                                       034770
      DO 126 J=1,NLINE                                                  034780
  126 TLINE(J)=BLANK                                                    034790
      DO 128 J=1,NLINE,10                                               034800
  128 TLINE(J) = COLON                                                  034810
C     PUT DOT ALONG THE X AXIS IF WE CAN                                034820
      IF(ONPAGE) TLINE(BIASP) = DOT                                     034830
      IF(SGNE) 130,130,132                                              034840
      IF(SGNE) 130,130,132                                              034850
```

```
  130  NFINAL=FIN/FMAX*BIAS+BIASP                                        034860
       GO TO 134                                                         034870
  132  NFINAL=FIN/FMAX*(BIASL-BIAS)+BIASP                                034880
  134  IF(NFINAL .GT. NLINE .OR. NFINAL .LT. 1) GO TO 136                034890
       TLINE(NFINAL) = DOT                                               034900
  136  DO 144 I=1,N                                                      034910
       FU=FT(T)                                                          034920
       IF(SGNE)140,140,138                                               034930
  138  NFU=FU/FMAX*(BIASL-BIAS)+BIASP                                    034940
       GO TO 142                                                         034950
  140  NFU=FU/FMAX*BIAS+BIASP                                            034960
  142  IF(NFU.GT.NLINE.OR.NFU.LT.1) NFU=102                              034970
       NTEMP = TLINE(NFU)                                                034980
       TLINE(NFU)=CROSS                                                  034990
       WRITE(NFILE,240)(TLINE(J),J=1,NLINE)                              035000
       TLINE(NFU) = NTEMP                                                035010
       T=T+DT                                                            035020
  144  CONTINUE                                                          035030
       GO TO 96                                                          035040
C                                                                        035050
C      SET UP FOR PRINTER PLOT                                           035060
C                                                                        035070
  146  NTOP = 200                                                        035080
       NFILE = 8                                                         035090
       NLINE = 101                                                       035100
       BIASL = 100.                                                      035110
       GO TO 108                                                         035120
C                                                                        035130
C      SET UP FOR CALCOMP PLOT                                           035140
C                                                                        035150
```

```
148   WRITE(7,285)                                                          035160
      READ *,T,DUR                                                          035170
      WRITE(7,255)                                                          035180
      READ(5,310) TITLE                                                     035190
      WRITE(7,275)                                                          035200
      READ *,FLET                                                           035210
      LETTERS = FLET                                                        035220
      DT=DUR/500.                                                           035230
      DO 150 J=1,501                                                        035240
      Y1(J)=FT(T)                                                           035250
      X(J)=T                                                                035260
150   T=T+DT                                                                035270
      CALL PLOT(0,-3.,-3)                                                   035280
      CALL PLOT(0,3.,-3)                                                    035290
      CALL SCALE(X,7.,501,1)                                                035300
      CALL SCALE(Y1,5.,501,1)                                               035310
      CALL AXIS(0,0,4HF(T),4,5.,90.,Y1(502),Y1(503))                        035320
      CALL AXIS(0,0,7HT (SEC),-7,7.,0,X(502),X(503))                        035330
      CALL LINE(X,Y1,501,1,0,0)                                             035340
      CALL SYMBOL((7.-LETTERS*.125)*.5,5.30,.125,TITLE,0,LETTERS)           035350
C                                                                           035360
C     DRAW 6" X 9" BOX AROUND PLOT                                          035370
C                                                                           035380
      CALL PLOT(-.5,-.5,3)                                                  035390
      CALL PLOT(7.5,-.5,2)                                                  035400
      CALL PLOT(7.5,5.5,2)                                                  035410
      CALL PLOT(-.5,5.5,2)                                                  035420
      CALL PLOT(-.5,-.5,2)                                                  035430
      CALL PLOT(10.,0,-3)                                                   035440
      GO TO 96                                                              035450
```

```
 200  FORMAT("-",12X,"ROOT",15X,"POWER",15X,"COEFFICIENT"/"0  REAL",12X,035460
C                                                                       035470
C     OUTPUT FORMATS                                                    035480
C                                                                       035490
     +  "IMAG",23X,"REAL",14X,"IMAG")                                   035500
 201  FORMAT("0  REAL",12X,"IMAG",21X,"MAGNITUDE",10X,"ANGLE")          035510
 202  FORMAT(G12.5,G15.5,I9,G19.5,F14.3)                                035520
 203  FORMAT(G12.5,G15.5,I9,G19.5,G17.5)                                035530
 204  FORMAT(G12.5,G15.5,8X,"1",G19.5,G17.5)                            035540
 205  FORMAT(G12.5,G15.5,8X,"1",G19.5,F14.3)                            035550
 223  FORMAT(".TYPE NO. OF ZEROS, POLES, POWER OF REPEATED ROOT, AND NUM035560
     +ERATOR CONSTANT.")                                               035570
 224  FORMAT(".TYPE REAL AND IMAG PARTS OF NUMERATOR FACTOR.")          035580
 225  FORMAT(".TYPE REAL AND IMAG PARTS OF DENOMINATOR FACTOR.")        035590
 226  FORMAT(".TYPE ONLY REAL PART OF REPEATED ROOT.")                  035600
 227  FORMAT("-CALCULATIONS COMPLETE"/" FOR TABULAR LISTING TYPE 1"/    035610
     +  " FOR TELETYPE PLOT TYPE 2" / " FOR PRINTER PLOT TYPE 3" /     035620
     +" FOR CALCOMP PLOT TYPE 4" /   " FOR ANOTHER PROBLEM TYPE 5" /   035630
     +" TO TERMINATE TYPE 0")                                          035640
 228  FORMAT(         "-THE TIME FUNCTION IS" /   " F(T)= ")            035650
 229  FORMAT(G19.5,"T**",I2,"  EXP(",G11.5,"T)")                        035660
 230  FORMAT(G19.5,"EXP(",G11.5,"T)")                                   035670
 231  FORMAT(G19.5,"EXP(",G11.5,"T) SIN(",G11.5,"*T+",F8.3,")")         035680
 234  FORMAT(".TYPE INITIAL TIME, PLOT DURATION, DELTA TIME, AND ORDINAT035690
     +E BIAS.")                                                        035700
 235  FORMAT(1X,2G16.8)                                                 035710
 237  FORMAT(G19.5,"T  EXP(",G11.5,"T)")                               035720
 240  FORMAT(1X,101A1)                                                  035730
 245  FORMAT("1",31X,"FINAL VALJE =",G15.8)                             035740
 246  FORMAT(3X,"TP =",G13.5)                                           035750
```

```
 243  FORMAT(8X,"T",15X,"F(T)")                                              035760
 249  FORMAT(32X,"PEAK VALUE =",G15.8)                                       035770
 250  FORMAT(".TYPE INITIAL TIME, LISTING TIME DURATION, DELTA TIME.")       035780
 255  FORMAT( " CONJUGATE ASSUMED")                                          035790
 260  FORMAT( ".TYPE OPTION")                                                035800
 265  FORMAT( " TYPE TITLE - 50 CHARACTERS MAX"/".",5(9X,","))               035810
 270  FORMAT( ".TYPE NO. OF CHARACTERS, INCL SPACES.")                       035820
 280  FORMAT(" ILLEGAL INPUT FOR NUMBER OF ROOTS OR POLES.")                 035830
 285  FORMAT(".TYPE INITIAL TIME, PLOT DURATION.")                           035840
 295  FORMAT("  GRAPH LIMITS ARE",G14.7," TO",G14.7)                         035850
 300  FORMAT(//" ================================================================035860
     +=================="//" ***************START OF PROGRAM 'PARTL'******035870
     +*********")                                                            035880
C                                                                           035890
C     INPUT FORMATS                                                         035900
C                                                                           035910
 310  FORMAT(5A10)                                                           035920
      END                                                                   035930


      SUBROUTINE POLAR                                                      035940
      COMMON /POLARC/ AC,BD,FACT,FACTR                                      035950
      DATA PI/3.1415926535/,HALFPI/1.5707963285/                            035960
      FACTR = SQRT(AC**2 + BD**2)                                           035970
      IF(AC) 6,1,5                                                          035980
1     IF(BD) 2,3,4                                                          035990
2     FACT = -HALFPI                                                        036000
      RETURN                                                                036010
```

```
7       FACT = 0                                                        036020
        RETURN                                                          036030
4       FACT = HALFPI                                                   036040
        RETURN                                                          036050
5       FACT = ATAN(BD/AC)                                              036060
        RETURN                                                          036070
6       IF(BD) 7,8,9                                                    036080
7       FACT = ATAN(BD/AC) - PI                                         036090
        RETURN                                                          036100
8       FACT = -PI                                                      036110
        RETURN                                                          036120
9       FACT = ATAN(BD/AC) + PI                                         036130
        RETURN                                                          036140
        END                                                             036150
```

```
        FUNCTION FT(T)                                                  036160
        COMMON Z(12),XR,NXX,Y(24),W(24),CR(24),EC(24),OM(24),FE(24),L   036170
        RAD = 3.1415926535/180.                                         036180
        IF(NXX.EQ.0)GO TO 2                                             036190
        EX=EXP(XR*T)                                                    036200
        F1 = 0                                                          036210
        DO 1  I=1,12                                                    036220
        F1 = F1 + Z(I)*EX                                               036230
1       EX = EX*T                                                       036240
        GO TO 3                                                         036250
2       F1=0.                                                           036260
3       F2=0.                                                           036270
```

```
      F3=0.                                                              036280
      DO 4   I=1,L                                                       036290
      F2=F2+Y(I)*EXP(W(I)*T)                                             036300
    4 F3=F3+CR(I)*EXP(EC(I)*T)*SIN(OM(I)*T+FE(I)* RAD)                   036310
      FT=F1+F2+F3                                                        036320
      RETURN                                                             036330
      END                                                                036340


      OVERLAY(CCPCSD,4,0)                                                036350
      PROGRAM POLY                                                       036360
C     ROOTS OF A NTH DEGREE POLYNOMIAL  (MULLER'S METHOD)                036370
      COMMON RR,RI,A(99),DA(99),DRR(99),DRI(99)                          036380
      DOUBLE PRECISION DA,DRR,DRI                                        036390
      COMPLEX CR                                                         036400
      EQUIVALENCE (CR,RR)                                                036410
      PRINT 890                                                          036420
  890 FORMAT(//" ================================================================036430
     +================"//" ***************START OF PROGRAM 'POLY'******036440
     +*********")                                                        036450
  211 WRITE(7,4)                                                         036460
    4 FORMAT("0TYPE N, DEGREE OF POLYNOMIAL, ON 1ST LINE, "              036470
     +"OR ENTER '0' TO STOP."/                                           036480
     -" THEN TYPE N+1 REAL COEFFICIENTS "                                036490
     -"(HIGHEST DEGREE COEFF. FIRST),"/                                  036500
     -" WITH UP TO 80 CHAR. PER LINE, NUMBERS SEPARATED BY COMMAS.")     036510
   40 WRITE(7,5)                                                         036520
    5 FORMAT("0? N= ")                                                   036530
```

```
      READ(5,*) N                                                          036540
      IF(N.LE.0.OR.EOF(5).NE.0) GO TO 500                                  036550
      IF(N.LT.99) GO TO 50                                                 036560
      WRITE(7,*) "N > 98. TRY AGAIN"                                       036570
      GO TO 40                                                             036580
   50 M=N+1                                                                036590
   45 WRITE(7,5)                                                           036600
    5 FORMAT(" ? A(I)= ")                                                  036610
      READ(5,*) (A(L),L=1,M)                                               036620
      AMAX=0.                                                              036630
      WRITE(7,105)                                                         036640
  105 FORMAT(*0 I *5X*A(I)*)                                               036650
      DO 101 L=1,M                                                         036660
      WRITE(7,110) L,A(L)                                                  036670
  110 FORMAT(1H I2,2X,G15.8)                                               036680
      DA(L)=A(L)                                                           036690
      IF(ABS(A(L)).GT.AMAX) AMAX=ABS(A(L))                                 036700
  101 CONTINUE                                                             036710
      DO 220 N1=1,M                                                        036720
      IF(A(N1).NE.0.) GO TO 225                                            036730
  220 CONTINUE                                                             036740
      WRITE(7,230)                                                         036750
  230 FORMAT(" ERROR...ALL COEFF'S ARE ZERO")                              036760
      GO TO 40                                                             036770
  225 NX=M-N1                                                              036780
      CALL DMULR(DA(N1),NX,DRR,DRI)                                        036790
      PRINT 1004                                                           036800
 1004 FORMAT(//" ***************************************************       036810
     +"/"    PLEASE COPY DOWN THE REAL AND IMAGINARY PARTS (THAT"/         036820
     +" WILL BE SHOWN) IN ROOT AND FACTOR FORM: I.E., ROOT...S=-2;"/       036830
```

```
     +" FACTOR...S+2."/                                                    035840
     +" ***************************************************               036850
     +"//)                                                                 036860
      WRITE(7,300)                                                         036870
  300 FORMAT(*0 I *2X*REAL:ROOT(I)* 4X*IMAG:ROOT(I)* 4X*RESIDUAL*)         036880
      DO 250 I=1,NX                                                        036890
      RR=ORR(I)                                                            036900
      RI=ORI(I)                                                            036910
      ERR=RESID(A(N1),NX,CR)/AMAX                                          036920
  250 WRITE(7,310) I,CR,ERR                                                036930
  310 FORMAT(1H I2,2X,G15.8,1X,G15.8,1X,1PE8.1)                            036940
      GO TO 40                                                             036950
  500 CONTINUE                                                             036960
      PRINT 501                                                            036970
  501 FORMAT(//"        --------END OF PROGRAM 'POLY'--------       "//)   036980
      END                                                                  036990


      FUNCTION RESID(A,N,X)                                                037000
      DIMENSION A(1)                                                       037010
      COMPLEX X,C                                                          037020
      C=A(1)                                                               037030
      DO 1 I=1,N                                                           037040
    1 C=C*X+A(I+1)                                                         037050
      RESID=CABS(C)                                                        037060
      RETURN                                                               037070
      END                                                                  037080
```

```
      SUBROUTINE DMULR (COE,N1,ROOTR,ROOTI)                              037090
C                                                                        037100
C                                                                        037110
C*************************************************************************037120
C                                                                        037130
C    POLYNOMIAL ROOT FINDER SUBROUTINE ....                              037140
C                                                                        037150
C    ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS ....                      037160
C                                                                        037170
C    THIS METHOD APPROXIMATES THE FUNCTION F(Z) BY A QUADRATIC           037180
C  WHICH MAY ,IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT         037190
C  REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(Z) THOUGH                  037200
C  THE FUNCTION F(Z) MUST BE EVALUATED ONCE PER ITERATION ....           037210
C                                                                        037220
C    THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL        037230
C  WITH REAL COEFFICIENTS ....                                           037240
C                                                                        037250
C                                                                        037260
C    USE OF MULLER SUBROUTINE ....                                       037270
C  1.  CALL DMULR  (COE,N1,ROOTR,ROOTI) ....                             037280
C  2.  COE IS THE TAG OF THE ARRAY OF COEFFICIENTS.                      037290
C      THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE              037300
C      TO LOWEST DEGREE .                                                037310
C  3.  N1 IS DEGREE OF THE POLYNOMIAL .                                  037320
C  4.  ROOTR IS THE TAG OF THE ARRAY WHERE THE REAL PARTS                037330
C      OF THE COMPLEX ROOTS ARE STORED .                                 037340
C  5.  ROOTI IS THE TAG OF THE ARRAY WHERE THE IMAGINARY                 037350
C      PARTS OF THE COMPLEX ROOTS ARE STORED ....                        037360
C                                                                        037370
C    ALL ARITHMETIC IS IN THE COMPLEX MODE ....                          037380
```

```
C     THEREFORE UNDER-FLOW IS NORMAL FOR REAL ROOTS ....                    037390
C                                                                           037400
C     MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE .                037410
C     WHEN MULTIPLICITY IS FOUR THE ACCURACY DECREASES TO                   037420
C     ABOUT TWO PLACES ....                                                 037430
C                                                                           037440
C     DEGREE SQUARED DIVIDED BY TWENTY ....                                 037450
C     FOR DEGREE ELEVEN IT TAKES SIX SECONDS ....                           037460
C                                                                           037470
C                                                                           037480
C                                                                           037490
C***************************************************************************037500
C                                                                           037510
C                                                                           037520
C                                                                           037530
      DOUBLE PRECISION ROOTR,ROOTI,AXR,AXI,ALP1R,ALP1I,TEM                  037540
      DOUBLE PRECISION BET1R,BET1I,ALP2R,ALP2I,BET2R,BET2I                  037550
      DOUBLE PRECISION TEMR,TEMI,ALP3R,ALP3I,BET3R,BET3I                    037560
      DOUBLE PRECISION ALP4R,ALP4I,TEM1,TEM2,HELL,BELL                      037570
      DOUBLE PRECISION TE1,TE2,TE3,TE4,TE5,TE6,TE7,TE8,TE9,TE10             037580
      DOUBLE PRECISION TE11,TE12,TE13,TE14,TE15,TE16,DE15,DE16,COE          037590
C                                                                           037600
      DIMENSION COE(1),ROOTR(1),ROOTI(1)                                    037610
C                                                                           037620
      N2=N1+1                                                               037630
      N4=0                                                                  037640
      I=N1+1                                                                037650
10    IF(COE(I))9,7,9                                                       037660
7     N4=N4+1                                                               037670
      ROOTR(N4)=0.000                                                       037680
```

```
      ROOTI(N4)=0.0D0                                                   037690
      I=I-1                                                             037700
      IF(N4-N1)19,37,19                                                 037710
9     CONTINUE                                                          037720
C                                                                       037730
10    AXR=0.8D0                                                         037740
      AXI=0.0D0                                                         037750
      L=1                                                               037760
      N3=1                                                              037770
      ALP1R=AXR                                                         037780
      ALP1I=AXI                                                         037790
      M=1                                                               037800
      GO TO 99                                                          037810
C                                                                       037820
11    BET1R=TEMR                                                        037830
      BET1I=TEMI                                                        037840
      AXR=0.85D0                                                        037850
      ALP2R=AXR                                                         037860
      ALP2I=AXI                                                         037870
      M=2                                                               037880
      GO TO 99                                                          037890
C                                                                       037900
12    BET2R=TEMR                                                        037910
      BET2I=TEMI                                                        037920
      AXR=0.9D0                                                         037930
      ALP3R=AXR                                                         037940
      ALP3I=AXI                                                         037950
      M=3                                                               037960
      GO TO 99                                                          037970
C                                                                       037980
```

```
13      BET3R=TE1R                                                        037990
        BET3I=TE1I                                                        038000
14      TE1=ALP1R-ALP3R                                                   038010
        TE2=ALP1I-ALP3I                                                   038020
        TE5=ALP3R-ALP2R                                                   038030
        TE6=ALP3I-ALP2I                                                   038040
        TEM=TE5*TE5+TE6*TE6                                               038050
        TE3=(TE1*TE5+TE2*TE6)/TEM                                         038060
        TE4=(TE2*TE5-TE1*TE6)/TEM                                         038070
        TE7=TE3+1.0D0                                                     038080
        TE9=TE3*TE3-TE4*TE4                                               038090
        TE10=2.0D0*TE3*TE4                                                038100
        DE15=TE7*BET3R-TE4*BET3I                                          038110
        DE16=TE7*BET3I+TE4*BET3R                                          038120
        TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15                               038130
        TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16                               038140
        TE7=TE9-1.0D0                                                     038150
        TE1=TE9*BET2R-TE10*BET2I                                          038160
        TE2=TE9*BET2I+TE10*BET2R                                          038170
        TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I                               038180
        TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R                               038190
        TE15=DE15*TE3-DE16*TE4                                            038200
        TE16=DE15*TE4+DE16*TE3                                            038210
        TE1=TE13*TE13-TE14*TE14-4.0D0*(TE11*TE15-TE12*TE16)               038220
        TE2=2.0D0*TE13*TE14-4.0D0*(TE12*TE15+TE11*TE16)                   038230
        TEM=DSQRT(TE1*TE1+TE2*TE2)                                        038240
        IF(TE1)113,113,112                                                038250
113     TE4=DSQRT(0.5D0*(TEM-TE1))                                        038260
        TE3=0.5D0*TE2/TE4                                                 038270
        GO TO 111                                                         038280
```

```
C                                                                       038290
112   TE3=DSQRT(0.500*(TE4+TE1))                                        038300
      IF(TE2)110,200,200                                                038310
110   TE3=-TE3                                                          038320
200   TE4=0.500*TE2/TE3                                                 038330
111   TE7=TE13+TE3                                                      038340
      TE8=TE14+TE4                                                      038350
      TE9=TE13-TE3                                                      038360
      TE10=TE14-TE4                                                     038370
      TE1=2.000*TE15                                                    038380
      TE2=2.000*TE16                                                    038390
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205                  038400
204   TE7=TE9                                                           038410
      TE8=TE10                                                          038420
205   TE4=TE7*TE7+TE8*TE8                                               038430
      TE3=(TE1*TE7+TE2*TE8)/TE4                                         038440
      TE4=(TE2*TE7-TE1*TE8)/TE4                                         038450
      AXR=ALP3R+TE3*TE5-TE4*TE6                                         038460
      AXI=ALP3I+TE3*TE6+TE4*TE5                                         038470
      ALP4R=AXR                                                         038480
      ALP4I=AXI                                                         038490
      M=4                                                               038500
      GO TO 99                                                          038510
C                                                                       038520
15    N5=1                                                              038530
C***********************************************************************038540
33    IF(DABS(HELL)+DABS(BELL)-1.0D-20)18,18,16                         038550
16    TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)                               038560
      IF(TE7/(DABS(AXR)+DABS(AXI))-1.0D-7)18,18,17                      038570
C***********************************************************************038580
```

```
17      N3=N3+1                                                          038590
        ALP1R=ALP2R                                                      038600
        ALP1I=ALP2I                                                      038610
        ALP2R=ALP3R                                                      038620
        ALP2I=ALP3I                                                      038630
        ALP3R=ALP4R                                                      038640
        ALP3I=ALP4I                                                      038650
        BET1R=BET2R                                                      038660
        BET1I=BET2I                                                      038670
        BET2R=BET3R                                                      038680
        BET2I=BET3I                                                      038690
        BET3R=TEMR                                                       038700
        BET3I=TEMI                                                       038710
        IF(N3-100)14,18,18                                               038720
18      N4=N4+1                                                          038730
        ROOTR(N4)=ALP4R                                                  038740
        ROOTI(N4)=ALP4I                                                  038750
        N3=0                                                             038760
41      IF(N4-N1)30,37,37                                                038770
37      RETURN                                                           038780
C*****************************************************************      038790
30      IF(DABS(ROOTI(N4))-1.0D-5)10,10,31                               038800
31      GO TO (32,10),L                                                  038810
32      AXR=ALP1R                                                        038820
        AXI=-ALP1I                                                       038830
        ALP1I=-ALP1I                                                     038840
        M=5                                                              038850
        GO TO 99                                                         038860
33      BET1R=TEMR                                                       038870
        BET1I=TEMI                                                       038880
```

```
      AXR=ALP2R                                                        038890
      AXI=-ALP2I                                                       038900
      ALP2I=-ALP2I                                                     038910
      M=6                                                              038920
      GO TO 99                                                         038930
C                                                                      038940
34    BET2R=TEMR                                                       038950
      BET2I=TEMI                                                       038960
      AXR=ALP3R                                                        038970
      AXI=-ALP3I                                                       038980
      ALP3I=-ALP3I                                                     038990
      L=2                                                              039000
      M=3                                                              039010
99    TEMR=COE(1)                                                      039020
      TEMI=0.000                                                       039030
      DO 100 I=1,N1                                                    039040
      TE1=TEMR*AXR-TEMI*AXI                                            039050
      TEMI=TEMI*AXR+TEMR*AXI                                           039060
100   TEMR=TE1+COE(I+1)                                                039070
      HELL=TEMR                                                        039080
      BELL=TEMI                                                        039090
42    IF(N4)102,103,102                                                039100
102   DO 101 I=1,N4                                                    039110
      TEM1=AXR-ROOTR(I)                                                039120
      TEM2=AXI-ROOTI(I)                                                039130
      TE1=TEM1*TEM1+TEM2*TEM2                                          039140
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1                                    039150
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1                                   039160
101   TEMR=TE2                                                         039170
103   GO TO (11,12,13,15,33,34),M                                      039180
      END
```

APPENDIX D

PROCEDURE FOR ADDING COMPUTER PROGRAMS (ALGORITHMS) TO THE CONSOLIDATED COMPUTER PROGRAM FOR CONTROL DESIGN (CCPCSD) (DECEMBER 1976)

Procedure for Adding Computer Programs (Algorithms) to the Consolidated Computer Program for Control System Design (CCPCSD)

The Consolidated Computer Program for Control System Design (CCPCSD) is structured to allow adding other computer programs without causing any substantial modification to the programs being added. The following procedure is prepared to help an individual add another computer program to the CCPCSD. Please read all of the procedure before starting.

(NOTE: Caution must be exercised before the following procedure is started as two major limitations exist.

a. The CCPCSD uses approximately $25{,}000_8$ words of core; the added program must not exceed $25{,}000_8$ words.

b. The maximum amount of files that can be used at the intercom terminal is ten.

Hence, make sure that your program does not cause the CCPCSD to exceed these limitations. Also, the steps mentioned below do not necessarily have to be done in the order given. And, further, for the purpose of this procedure, it is assumed that the program to be added is called "PROJX," and that it is the fifth program in the CCPCSD.)

STEP 1. IMPORTANT: PROJX must be capable of fully independent operation from an intercom terminal. (All of the errors must have been removed before adding the program to the CCPCSD.)

STEP 2. Compare all of the file names listed in the PROGRAM statements of the CCPCSD and PROJX. PROJX must now be altered, if necessary, to agree with the CCPCSD file names.

EXAMPLE: The PROGRAM statement for the CCPCSD is:

```
PROGRAM CCPCSD(INPUT=100B,OUTPUT=100B,ANSWER,PPLOT,
CFILE,PLOT,TAPE5=INPUT,TAPE6=ANSWER,TAPE7=OUTPUT,
TAPE8=PPLOT,TAPE9=100B).
```

Assume that the PROJX PROGRAM statement is:

```
PROGRAM PROJX(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=
OUTPUT,COUNT).
```

a. All of the statements in the PROJX program that refer to TAPE1 must be changed to refer to TAPE5; and, consequently, TAPE2 changed to TAPE7.

From	To
READ(1,200)	READ(5,200)
WRITE(2,210)	WRITE(7,210)
ETC.	

b. Make a new PROGRAM statement with the CCPCSD files corrections:

```
PROGRAM PROJX(INPUT,OUTPUT,TAPE5=INPUT,TAPE7=
OUTPUT,COUNT).
```

STEP 3. This step is very important, and could get rather tedious. If COMMON statements are used in PROJX, all of the file names for the COMMON statements in programs ROOTL, PARTL, FREQR, and POLY must be compared with those in PROJX. If any file names are identical, modify the PROJX COMMON statements

by using new file names. (Do not forget to make the appropriate changes throughout program PROJX to reflect the new file names.)

If PROJX does not have any COMMON statements, you may proceed to STEP 4.

STEP 4. Now that you have finished the modifications for the COMMON statements and the PROGRAM statements throughout PROJX, test the program to insure that it still functions properly. Debug, if necessary.

STEP 5. Remove the PROGRAM statement from PROJX and substitute the following statements in its place:

```
OVERLAY(CCPCSD,5,0)
PROGRAM PROJX
```

Now you are ready to make changes to the executive overlay (Overlay(CCPCSD,0,0)) of the CCPCSD.

STEP 6. (Reference STEP 2) - The file named COUNT, shown in the PROGRAM statement for PROJX, must be added to the PROGRAM statement for the executive overlay of the CCPCSD (i.e., ". . .TAPE9=100B, COUNT)").

STEP 7. To the FORMAT statement 1005 (approximately line number 730), add the symbolic name PROJX, and an explanation of PROJX (similar to those given for ROOTL and PARTL).

STEP 8. Statement 18 of the executive overlay reads

"18 IF(J.EQ.5HPARTL) GO TO 9." Insert the statement "IF(J.EQ.5HPROJX) GO TO 21" after statement 18.

STEP 9. Just before the END statement (approximately line number 1110), add the following statements:

```
21   CALL OVERLAY(CCPCSD,5,0,0)
     GO TO 10
```

STEP 10. Physically put the cards for program PROJX behind (at the end) of the cards in the CCPCSD. Hence, the new structure would be (in general):

```
OVERLAY(CCPCSD,0,0)        Executive
OVERLAY(CCPCSD,1,0)        ROOTL
OVERLAY(CCPCSD,2,0)        FREQR
OVERLAY(CCPCSD,3,0)        PARTL
OVERLAY(CCPCSD,4,0)        POLY
OVERLAY(CCPCSD,5,0)        PROJX
```

STEP 11. Put the new CCPCSD into the permanent files, or library. Test the CCPCSD at the intercom terminal. Debug, if necessary. Re-catalog, if necessary.

* *

This procedure for adding programs to the CCPCSD is developed to be relatively simple and easy to understand. However, some of the changes that must be made to the CCPCSD and to the program being added should be made with care. Typing the changes incorrectly can cause many unnecessary headaches.

Good luck.

VITA

Frederic L. O'Brien was born on 20 November 1943 in Ilion, New York, the son of Frederick B. O'Brien and Helen M. O'Brien. Six Months after graduating from Zephyrhills High School, Zephyrhills, Florida, in 1961, he entered the United States Air Force as an Airman Basic. In 1967 he began the Airman Education and Commissioning Program (AECP) at Arizona State University, Tempe, Arizona. He graduated in January of 1971, and, after completing Officer Training School, he was commissioned as a Second Lieutenant. He spent four years at Griffiss Air Force Base, New York, as a Project Engineer for a Rome Air Development Center (RADC) program. He entered the Air Force Institute of Technology in May 1975 with fourteen years of military service.

Permanent Address: 4299 West Humphrey Street
Tampa, Florida 33614

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GE/EE/76D-38	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A CONSOLIDATED COMPUTER PROGRAM FOR CONTROL SYSTEM DESIGN (CCPCSD)		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Frederic L. O'Brien Capt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology(AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory(AFFDL) Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1976
		13. NUMBER OF PAGES 220
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Jerrel F. Guess, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Consolidated Computer Program
Combined Program
Root Locus, Frequency Response, Time Response Programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several different computer programs have been independently developed for control system design and analysis. Each program may have its own input and output formats (different from any other), and each program may have its own operating philosophy. This report describes a Consolidated Computer Program for Control System Design (CCPCSD) and its development, specifications, organization, realization, testing, and results. (cont on p 1473B)

DD FORM 1 JAN 73 1473 A EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The CCPCSD contains four previously written Air Force Institute of Technology (AFIT) computer programs for control system design algorithms (ROOTL, FREQR, PARTL, and POLY) and a lead-in program which controls the overall flow of the CCPCSD. An overlay structure is used. The structure consists principally of an executive (main) overlay and four primary overlays (one for each subprogram mentioned above). Each subprogram is modified towards standardizing inputs, termination procedures, and data structure (where feasible).

The CCPCSD is developed to be used by control systems engineers and/or students. A user-oriented approach permeates the entire program. This computer-aided design tool can be operated from an intercom terminal, where operator-computer interaction takes place.